

1

AD-A230 468



DTIC  
ELECTE  
JAN 3 1991

D

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY  
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

91 1 3 135

AFIT/GE/ENG/90D-04

BRIGHTNESS INVARIANT PORT  
RECOGNITION FOR  
ROBOTIC AIRCRAFT REFUELING

THESIS

Richard A. Bennett  
Capt USAF

AFIT/GE/ENG/90D-04

Approved for public release; distribution unlimited

BRIGHTNESS INVARIANT PORT RECOGNITION  
FOR ROBOTIC AIRCRAFT REFUELING

THESIS

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University  
In Partial Fulfillment of the  
Requirements for the Degree of  
Master of Science in Electrical Engineering

Richard A. Bennett, B.S.E.E.

Capt USAF

December 13, 1990

Accession For	
NTIS GPARI	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited

## *Preface*

I would like to sincerely thank my thesis advisor Dr. Michael Leahy for giving me complete autonomy during this research to go the direction I felt the vision system needed to go, and his support, help, and guidance in achieving the final product. I would also like to thank Dr. Y. C. Shiu from Wright State University, for his tremendous help in describing how to perform edge detection and for his patience, putting up with my questions and problems when I couldn't get the code to cooperate. I would like to thank Dan Zambon for keeping all the systems running.

Finally, I would like to thank my family for putting up with me for the last 18 months. I would like to thank my daughters, Elizabeth and Laura, for their understanding when I couldn't help them with their homework, and my sons, Patrick and Jeremy, for their understanding when I tied up the computer and they couldn't play their computer games. I especially want to thank my wife, Kathleen, for her patience and understanding for all the things that didn't get done for the last 18 months. Without her love and support, I would not have made it through this program.

Richard A. Bennett

## *Table of Contents*

	<b>Page</b>
<b>Preface</b> . . . . .	<b>ii</b>
<b>Table of Contents</b> . . . . .	<b>iii</b>
<b>List of Figures</b> . . . . .	<b>vii</b>
<b>Abstract</b> . . . . .	<b>x</b>
 <b>I. Introduction</b> . . . . .	 <b>1-1</b>
1.1 Motivation . . . . .	1-1
1.2 Problem Statement . . . . .	1-2
1.3 Objective . . . . .	1-2
1.4 Background . . . . .	1-2
1.5 Equipment . . . . .	1-4
1.6 Contribution . . . . .	1-5
1.7 Organization . . . . .	1-6
 <b>II. Literature Review</b> . . . . .	 <b>2-1</b>
2.1 Introduction . . . . .	2-1
2.2 Edge Detection . . . . .	2-1
2.3 Object Recognition Techniques . . . . .	2-5
2.3.1 Template Matching. . . . .	2-5
2.3.2 High Level Knowledge Representation . . . . .	2-6
2.4 Image Enhancement . . . . .	2-6
2.4.1 Histogram Adjustment . . . . .	2-6
2.4.2 Fourier Transforms . . . . .	2-7
2.4.3 Brightness Normalization . . . . .	2-8

	Page
2.4.4 Homomorphic Image Filtering . . . . .	2-9
2.5 Object Recognition Implementations . . . . .	2-10
2.5.1 AFIT Distortion Invariant Algorithm . . . . .	2-10
2.5.2 Autonomous Face Recognition Machine (AFRM) . .	2-13
2.5.3 Robotic Visual Servoing System (RVSS) . . . . .	2-13
2.5.4 Fuel Interchange System . . . . .	2-15
2.6 Summary . . . . .	2-16
III. Refueling Port Recognition . . . . .	3-1
3.1 Introduction . . . . .	3-1
3.2 Initial Processing . . . . .	3-2
3.2.1 Image Acquisition . . . . .	3-2
3.2.2 Sobel Edge Detection . . . . .	3-4
3.2.3 Non-maximal Suppression . . . . .	3-4
3.2.4 Edge Linked-List Formation . . . . .	3-7
3.3 Edge Processing . . . . .	3-9
3.3.1 Line Merging . . . . .	3-9
3.3.2 Closed Loop Formation . . . . .	3-9
3.4 Loop Identification . . . . .	3-10
3.5 Knowledge Driven Port Recognition . . . . .	3-12
3.6 Searching for the Port . . . . .	3-13
3.7 Summary . . . . .	3-18
IV. Results and Discussion . . . . .	4-1
4.1 Introduction . . . . .	4-1
4.2 Initial Processing. . . . .	4-2
4.2.1 Non-maximal Suppression . . . . .	4-3
4.2.2 Thresholding . . . . .	4-3

	Page
4.3 Edge Processing . . . . .	4-5
4.4 Port Recognition . . . . .	4-6
4.4.1 Brightness Invariance . . . . .	4-7
4.4.2 Rotation Invariance . . . . .	4-7
4.4.3 Size Invariance . . . . .	4-7
4.4.4 Recognition with Partial Indicators . . . . .	4-11
4.5 Searching for the Port . . . . .	4-16
4.6 Limitations and Problems . . . . .	4-18
4.7 Conclusion . . . . .	4-21
 V. Conclusions and Recommendations. . . . .	 5-1
5.1 Summary. . . . .	5-1
5.1.1 Port Recognition. . . . .	5-2
5.1.2 Light Invariance. . . . .	5-2
5.1.3 Partial Indicators. . . . .	5-2
5.2 Conclusion. . . . .	5-2
5.3 Recommendations. . . . .	5-3
5.3.1 Thresholding. . . . .	5-3
5.3.2 Vaccelerator Board Usage. . . . .	5-3
5.3.3 ITEX Image Processing Board. . . . .	5-3
5.3.4 Edge Processing. . . . .	5-3
5.3.5 Camera Calibration. . . . .	5-3
5.3.6 Dynamic Tracking. . . . .	5-3
 Appendix A. Brightness Invariant Port Recognition System (BIPRS) Soft- ware Flow . . . . .	 A-1
A.1 Introduction . . . . .	A-1
 Bibliography . . . . .	 BIB-1

	Page
Vita . . . . .	VITA-1



## *List of Figures*

Figure	Page
1.1. Example of a Proposed Ground-Based Refueling System. . . . .	1-3
1.2. Example of Miller's White Circle on Black Background. . . . .	1-4
1.3. Example of Shipman's Port Indicators. . . . .	1-5
2.1. Intensity Edges a) Step b) Ramp c) Roof. . . . .	2-2
2.2. Example of Image and Corresponding Histogram. . . . .	2-7
2.3. Example of a Brightness Normalized Image. a) Original b) Normalized	2-9
2.4. Processing Steps in Homomorphic Image Filtering. . . . .	2-11
2.5. Example Homomorphic Filtering a) Original b) Filtered. . . . .	2-11
2.6. The AFIT Distortion Invariant Pattern Recognition Algorithm. . . . .	2-12
2.7. Refueling Port with Indicators. . . . .	2-14
2.8. Relationship of Indicator Parameters. . . . .	2-14
2.9. Refueling Receptacle with Point Targets. . . . .	2-15
3.1. Refueling Port with Rectangular Indicators. . . . .	3-2
3.2. Video Coordinate System. . . . .	3-3
3.3. Example of an Image Edge. . . . .	3-5
3.4. Example of a Sobel Operator Edge. . . . .	3-6
3.5. Sobel Edge Operator Directions. . . . .	3-6
3.6. Strength and Direction Bit Storage. . . . .	3-7
3.7. Example of Sobel Edge Strength Distribution. . . . .	3-8
3.8. Non-Maximum Suppression Thinned Sobel Edge. . . . .	3-8
3.9. Example of Edge Combination. . . . .	3-10
3.10. Shape Approximation Process for Closed Loops. . . . .	3-11
3.11. Indicator Parameter Relationships. . . . .	3-14

Figure	Page
3.12. Search Pattern. . . . .	3-15
3.13. Image With Only Two Indicators Visible. . . . .	3-16
3.14. Image With Two Indicators And a Partially Visible Third Indicator. . .	3-16
3.15. Image With All Three Indicators Visible. . . . .	3-17
3.16. Image With Top And Two Partial Indicators Visible. . . . .	3-17
3.17. Image With Two Indicators And a Partially Visible Third Indicator. . .	3-18
4.1. Original Image of the Refueling Port. . . . .	4-2
4.2. Example of Sobel Edge Image. . . . .	4-3
4.3. Non-Maximal Suppression Image Without Thresholding. . . . .	4-4
4.4. Non-Maximal Suppression Image With Thresholding. . . . .	4-4
4.5. Non-Maximal Suppression Image Without Thresholding Dark Lighting.	4-5
4.6. Non-Maximal Suppression Image With Thresholding Dark Lighting. . .	4-6
4.7. Original Image of the Refueling Port (Normal Room Light). . . . .	4-8
4.8. Non-Maximal Suppression Image of the Refueling Port Image (Normal Room Light). . . . .	4-8
4.9. Image with Limited Light. . . . .	4-9
4.10. Non-Maximal Suppression of Image with Limited Light. . . . .	4-9
4.11. Image with Bright Light. . . . .	4-10
4.12. Non-Maximal Suppression Image with Bright Light. . . . .	4-10
4.13. Rotated Image of the Refueling Port. . . . .	4-11
4.14. Image of the Refueling Port (Close). . . . .	4-12
4.15. Image of the Refueling Port (Far). . . . .	4-12
4.16. Top with Two Partial Indicators Present. . . . .	4-13
4.17. Image with Two Indicators. . . . .	4-14
4.18. Partial Top and Side Indicators. . . . .	4-14
4.19. Top Indicator Plus Right Side Indicator. . . . .	4-15
4.20. Top Indicator Plus Left Side Indicator. . . . .	4-15

Figure	Page
4.21. Initial Position for the Search for the Port. . . . .	4-16
4.22. Search Image with Two Indicators Present. . . . .	4-17
4.23. Image with Two Indicators Centered. . . . .	4-17
4.24. Final Image with All Indicators Visible. . . . .	4-19
4.25. Final Indicator Image. . . . .	4-19
4.26. Final Position for Refueling. . . . .	4-20
4.27. Closeup of Final Position for Refueling. . . . .	4-20
 A.1. Brightness Invariant Port Recognition System (BIPRS) Logic Flow . .	 A-2
A.2. <i>Sobelmain</i> Logic Flow . . . . .	A-3
A.3. <i>NMSmain</i> Logic Flow . . . . .	A-4
A.4. <i>Linkmain</i> Logic Flow . . . . .	A-5
A.5. <i>Searchmain</i> Logic Flow . . . . .	A-6
A.6. Non-Maximal Suppression Logic Diagram . . . . .	A-7
A.7. <i>Edges</i> Logic Diagram . . . . .	A-8
A.8. <i>Comblist</i> Logic Diagram . . . . .	A-9
A.9. <i>Findport</i> Logic Diagram . . . . .	A-10

*Abstract*

The Air Force is interested in the development of autonomous robotic systems for many possible military applications. One such application that the robotics research group at the Air Force Institute of Technology (AFIT) is actively researching is autonomous aircraft refueling. The approach being studied at AFIT involves the interfacing of computer vision capabilities with robot compliant motion technology. The vision system is used to locate and identify the refueling port and then passes the port location to a trajectory planner. The trajectory planner computes the movement necessary to move the robot refueling arm to the refueling port. After the robot arm is moved into close proximity of the refueling port, compliant motion techniques are used to insert the refueling nozzle into the port.

The purpose of this thesis effort was to develop and test a brightness invariant recognition algorithm that would locate and identify the half-scale mock-up of a Universal Aerial Refueling Receptacle Slipway Installation (UARRSI) aerial refueling port in different lighting conditions. This was accomplished by developing a brightness invariant port recognition system which relied on edge detection, line merging, loop formation, and knowledge driven recognition of identifiable attributes of the refueling port to locate and identify the UARRSI port.

Based on this approach, a brightness invariant port recognition system (BIPRS) was demonstrated which identified the refueling receptacle in different lighting conditions. The BIPRS was also invariant to orientation and size of the UARRSI port. The BIPRS demonstrated the feasibility of the autonomous aircraft refueling task, and provides an incentive for additional research in autonomous robot applications.

# BRIGHTNESS INVARIANT PORT RECOGNITION FOR ROBOTIC AIRCRAFT REFUELING

## *I. Introduction*

### *1.1 Motivation*

Currently, the development of autonomous robotic systems is a major interest to the Air Force. With the reduction of resources and personnel that the Air Force is facing in the future, the Air Force will have to learn to do more with less. To better use the Air Force's resources and to help alleviate or eliminate manpower intensive activities, it is becoming more critical to use autonomous robotic systems to perform some logistic and maintenance functions. Some activities that autonomous robotic systems can be used for are; depot maintenance, material distribution, munitions handling, rapid runway repair, and aircraft servicing [14]. All these activities require the robotic system to use sensors to find its position in relation to its environment and to find the object the robot is to work on. One type of sensor that will be used extensively is machine vision.

Machine/Computer vision is an important feedback sensor that gives a robotic system information about its environment. Computer vision can identify and locate objects so that the robot can perform some operation on them. The vision system can provide accurate position information without the use of other sensors or provide only approximate position information and rely on other sensors to provide close in information.

One application of computer vision that the Air Force Institute of Technology (AFIT) has been investigating is robotic aircraft refueling. For the robotic refueling system concept being researched at AFIT, the vision system provides rough position information on the location of the refueling port to the robot controller, and compliant motion techniques are used to perform the actual insertion of the nozzle into the port [15:1145].

## *1.2 Problem Statement*

In order to conduct autonomous aircraft refueling, a brightness invariant port recognition algorithm must be developed to effectively recognize the refueling port in a wide range of lighting conditions. An effective recognition system must function in different lighting conditions since ideal lighting, without shadows, can not always be guaranteed.

## *1.3 Objective*

The objective of this thesis effort was to develop a vision recognition algorithm and demonstrate the concept of light invariant object recognition for autonomous robotic refueling of aircraft. In order to successfully demonstrate an efficient port recognition function the following areas were addressed:

- Development of an algorithm which recognizes the refueling port with a minimum of false alarms,
- Recognition of the refueling port when only part of the port is visible in the image,
- Recognition over a wide range of lighting conditions,
- A more efficient search algorithm to locate the refueling port.

## *1.4 Background*

Currently, ground-based aircraft refueling requires several individuals to complete the complex task. The ground refueling task requires three well trained personnel to perform the grounding of the aircraft and connecting the refueling nozzle to the side of the aircraft. The task is more difficult to perform in a hostile, chemical or biological environment [15:1145]. One proposal for a ground-based autonomous robotic refueling is to drive a refueling vehicle close to the aircraft and have a robotic arm reach over the aircraft to the aerial refueling port to complete the refueling as shown in Figure 1.1 [22:1.3]. With this approach, one individual could sit safely in the cab of the truck and would not have to be exposed to a potentially hostile environment to refuel aircraft.

Aerial aircraft refueling is also a complex task that requires well trained personnel to manually guide the refueling boom into position and insert the refueling nozzle into the fuel port [22:1.1]. One proposal for a robotic aerial refueling scheme would have the refueling personnel place the refueling boom in close proximity to the aircraft to be fueled and then let the robotic refueler locate the refueling port and accomplish the insertion and actual refueling.

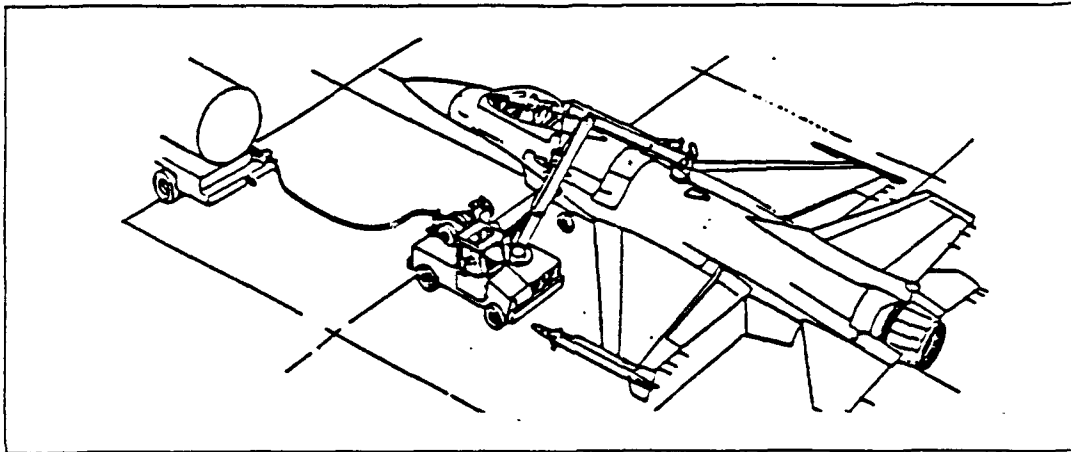


Figure 1.1. Example of a Proposed Ground-Based Refueling System [22:1.3].

The Air Force Institute of Technology (AFIT) has been actively investigating methods for recognizing a refueling port, visually servoing the robot arm to the refueling port, and for visually tracking the port. In 1987, Capt Miller used a white circle on a black background (see Figure 1.2) and demonstrated that visually locating an object and guiding the robot manipulator to an object was feasible [16, 22]. Miller's vision algorithm could not track the circle if it moved and was very inefficient, but it showed that vision can be used to acquire the refueling port for robotic refueling. In 1988, Capt Salisbury attempted to investigate the visual tracking of a white circle on a black background using a Kalman filter to predict and update the circle's position, but didn't complete his research [22:1.4]. In 1989, Capt Shipman demonstrated a static look-and-move robotic visual servoing system (RVSS) which identified, located, and servoed the PUMA 560's end effector to the slipway of a simulated half scale refueling port [22:1.2]. Shipman's port recognition algorithm relied on white passive indicator markings (see Figure 1.3) placed around the refueling port to

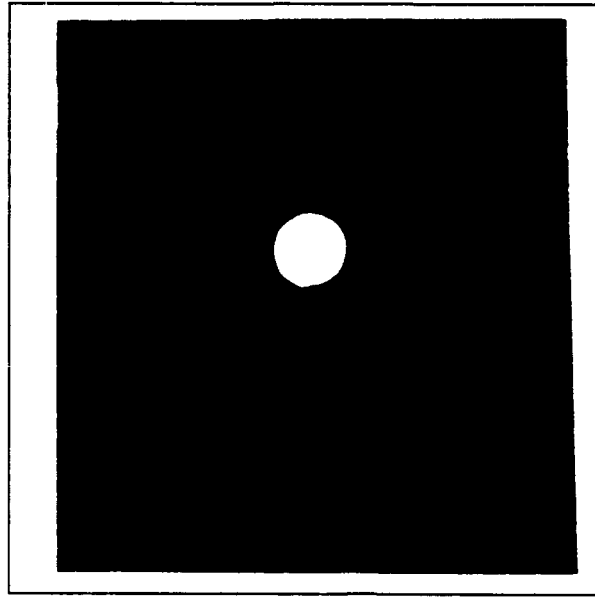


Figure 1.2. Example of Miller's White Circle on Black Background [22:3.4].

highlight the port. The RVSS system required ideal lighting conditions to identify the port. The port recognition process failed when the refueling port was placed in shadows.

AFIT has also been researching compliant motion techniques to complete the insertion of the refueling nozzle into the refueling port [1]. The compliant motion relies on a force sensor on the end of link three of the robot to feedback information on the contact forces encountered during the insertion of the nozzle. This information is used to adjust the trajectory of the robot arm to reduce the force encountered and still reach the target point in the trajectory.

### *1.5 Equipment*

The robotic systems laboratory provided the experimental environment necessary to investigate robotic refueling. The specific resources used to support this research were:

- PUMA 560 six degree-of-freedom robot manipulator
- ITEX 100 image processing system
- VAXstation III microcomputer



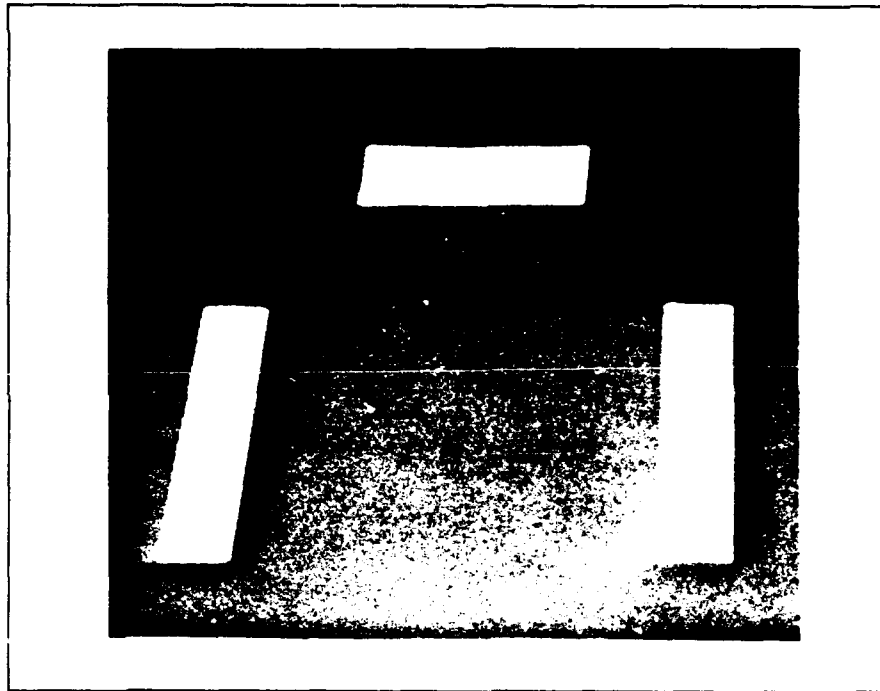


Figure 1.3. Example of Shipman's Port Indicators [22:3.11].

- Sony compact CCD Video Camera
- Half scale mock-up of a Universal Aerial Refueling Receptacle Slipway Installation (UARSSI) aircraft refueling port.

### *1.6 Contribution*

A brightness invariant object recognition algorithm was developed and demonstrated. The object recognition algorithm was able to identify the refueling port at different distances and orientations. Improvements in the recognition algorithm allow detection of the refueling port with only half of the port visible in the image and permit a more efficient port search strategy. The new algorithm provides AFIT with the visual recognition capability required to demonstrate the concept of autonomous robotic refueling under realistic unstructured lighting conditions.

### **1.7 Organization**

This thesis is organized as follows: Chapter 2 presents a review of current literature in the areas of object recognition, image normalization, edge detection, and visual servoing. Chapter 2 also provides a review of some of the past research efforts at AFIT such as the AFIT Distortion Invariant Pattern Recognition Algorithm, the Autonomous Face Recognizer Machine, and the Robotic Visual Servoing System. Chapter 3 contains a detailed description of the recognition algorithm theory and implementation used in this research effort. Chapter 4 provides a detailed analysis of the results from algorithm application and the problems that were encountered. Limitations encountered with the algorithm's implementation are also discussed. Chapter 5 presents the conclusions and provides recommendations for future refueling research. Appendix A presents the logic diagrams for the software developed during this research effort. Detailed software documentation is contained in a Robotic Systems Laboratory Internal Report [18].

## *II. Literature Review*

### *2.1 Introduction*

The design and implementation of an autonomous robot aircraft refueling system is a very complex task. The task requires interfacing a dynamic vision system with robotic control techniques. The vision system must locate and identify the refueling port under a wide range of lighting conditions and send the location of the port to the trajectory planner. The vision system must perform its function in near real time to provide adequate position updates to the trajectory planner so that tracking commands can be sent to the robot controller. The robot controller will then move the refueling arm into position to refuel the aircraft.

A review of current efforts in image processing and object recognition was conducted to determine the state of the art of techniques, and to select the approach that would be taken to solve the problem of identifying and locating the refueling port. Image processing was reviewed to identify ways to improve the quality of processed images. Object recognition was reviewed to identify techniques to isolate objects in an image and to highlight higher level recognition techniques that can be used to identify the refueling port in the image. The specific areas that were reviewed includes image brightness normalization, edge detection, and object recognition techniques. The selection of these topics was not intended to be all encompassing in the area of image processing and object recognition, but to provide an overview of the object recognition process.

### *2.2 Edge Detection*

A major problem in image processing for object recognition is the isolation and classification of objects in the image. The detection of an object's edges in the image is important to separate the object from other items in an image. Edge detection is the process that identifies the intensity or texture changes in an image. Object recognition combines the edge information with higher level knowledge representations to identify objects [25:147]. Edges can be classified as either intensity edges or texture edges [8:1338]. Intensity edges are easier to detect than texture edges since they are characterized by abrupt intensity

changes. Intensity edges can be either step, ramp, or roof edges (Figure 2.1). Intensity edges are the most common type of edge encountered and occur at changes in color in an image or at the edges of individual objects.

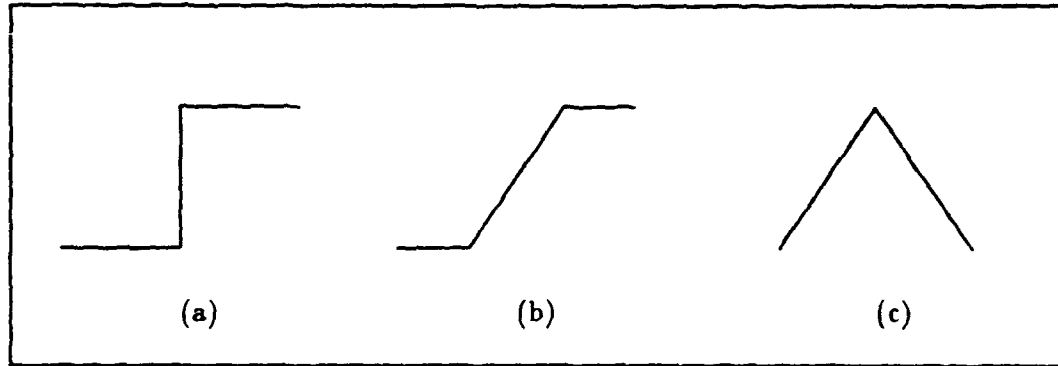


Figure 2.1. Intensity Edges a) Step b) Ramp c) Roof [8:1338].

Eom and Kashyap presented the steps needed to identify potential edge pixels [10:81-93]. Their approach involves taking the directional derivatives with the first order directional derivative shown in Equation 2.1 and the second order directional derivative shown in Equation 2.2 [10:82-85]. Eom and Kashyap's edge detection algorithm combines

$$\frac{\partial g}{\partial \alpha} = S(x, y) = \frac{\partial g}{\partial x} \cos \alpha + \frac{\partial g}{\partial y} \sin \alpha \quad (2.1)$$

$$\frac{\partial^2 g}{\partial \alpha^2} = \frac{\partial^2 g}{\partial x^2} \cos^2 \alpha + 2 \frac{\partial^2 g}{\partial x \partial y} \cos \alpha \sin \alpha + \frac{\partial^2 g}{\partial y^2} \sin^2 \alpha \quad (2.2)$$

conventional edge detection techniques that are good for detecting intensity differences, with an additional tests to detect texture changes. The first step in Eom and Kashyap's edge hypothesis generation is to find all the pixels that are potential edge pixels. Eom and Kashyap employ the Prewitt edge operators  $D_x$  and  $D_y$  to detect edges in the image [10:85].

$$D_x = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$D_y = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

The function  $g(x, y)$  is defined as the image intensity at location  $(x, y)$ .  $D_x$  and  $D_y$  are convolved with an image  $g(x, y)$  to form  $\frac{\partial g}{\partial x}$  and  $\frac{\partial g}{\partial y}$ . The angle of gradient direction  $\alpha$  is shown in Equation 2.3 and gives the direction that the edge faces in the image.

$$\alpha = \tan^{-1} \frac{\partial g / \partial y}{\partial g / \partial x} \quad (2.3)$$

A pixel with a first order directional derivative larger than the threshold value and a negative second order directional derivative is classified as a potential edge pixel [10:84]. The possible edges identified by directional derivatives could be intensity edges, texture edges, or spurious edges. The second step in the edge detection is to confirm the existence of edges. The quantized edge direction information is used for confirmation tests to verify that the pixel is an actual edge pixel. A likelihood ratio test with statistical texture modeling is used to test for texture edges and a difference of weighted averages is used to test for intensity edges [10:82-83]. Once the edges of the objects in the image have been identified, the information is passed to higher level algorithms to identify specific objects in the image.

There are many variants of the edge detection operator (Prewitt operator) used by Eom and Kashyap to locate edges in an image. All these operators use masks that are convolved with the target image to identify pixels and their spatial gradients [19:146-160].

Two such operators are the Roberts "cross" operators

$$D_y(\text{Roberts}) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.4)$$

$$D_x(\text{Roberts}) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2.5)$$

and the Sobel edge operators.

$$D_x(\text{Sobel}) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$D_y(\text{Sobel}) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.7)$$

Both the Roberts and Sobel operators are convolved with the image  $g(x, y)$  to obtain relative strengths ( $S_x$  and  $S_y$ ) in the  $x$  and  $y$  directions for each pixel location in  $g(x, y)$ .

$$S_x(x, y) = \sum_{i,j=1}^3 g(x+i, y+j) D_x(i, j) \quad (2.8)$$

$$S_y(x, y) = \sum_{i,j=1}^3 g(x+i, y+j) D_y(i, j) \quad (2.9)$$

where  $i$  and  $j$  represent the row and column of the edge operator used. The overall strength  $S(x, y)$  and a direction,  $\alpha$  for each pixel location in the image are given by [5:76-80].

$$S(x, y) = (S_x^2 + S_y^2)^{1/2} \quad (2.10)$$

$$\alpha = \tan^{-1}(S_y/S_x) \quad (2.11)$$

Roberts is one of the first edge operators and provides a good approximation of local changes in an image. The Sobel and Prewitt operators perform better than the Robert's operator by using local averaging to reduce the effects of noise on the image [5:77]. Both the Roberts and Sobel operators are implemented on the ITEX image processing board.

## 2.3 Object Recognition Techniques

Numerous object recognition techniques have been used in attempts to efficiently identify objects in a scene. Most of those techniques perform some kind of template matching and depend, to some extent, on the detection of edges to isolate individual objects in the image [24:699].

*2.3.1 Template Matching.* Template matching is a simple technique to detect specific features in an image. Most pattern recognition algorithms perform template matching using correlation, by sliding the template over the image looking for a match [23:63]. One measure of the similarity between an image and a template is Euclidean distance  $d(y)^2$  given by

$$d(y)^2 = \sum_x (f(x) - t(x - y))^2 \quad (2.12)$$

where  $f(x)$  is the image and  $t(x - y)$  is the template [5:66].

If the template is identical to the image being matched, then the distance is zero, a perfect match. If the template and the image don't match exactly, then the value of  $d(y)^2$  will be greater than zero and less than one. The success of the match will be based on the probability of a match given the template and the information available in the image. Template matching works well when the object in the image is the same size as the object in the template. If there is a large difference in size or orientation the template matching techniques fails. If the template is large, one should perform the template matching in the frequency domain by performing a Fourier transform of the image.

**2.3.2 High Level Knowledge Representation** High level knowledge representation (HLKR) is the process of using identifiable attributes of an object to perform object recognition. HLHR usually entails using some level of artificial intelligence to piece together attributes of an object identified in an image with a data base of objects and their attributes. If there is a large data base of possible objects, the search for an object that matches the attributes can be time consuming. If the data base of objects is small, the data base search can be very efficient. Looking for identifiable attributes can be a very reasonable approach for identifying objects of interest. For example consider a data base with only two objects of interest, a truck and a tank. An image contains an object with a small rectangular shape on top of a large rectangular shape and there is a long cylindrical shape protruding out of the small rectangle. When the object in the image is compared with the attributes of the two objects in the data base, the object would be identified as a tank. [21:3-10]

## **2.4 Image Enhancement**

One problem that hinders pattern recognition is the variance of lighting from one image to another. Shadows can hide edges or create false edges in an image changing the appearance of an object. Comparing a template made under ideal lighting with the same object in a shadow or bright light could result in no match between the image and the template. The affect of variations in lighting can be minimized by enhancing the image to highlight the important details. Some of the techniques that can be used to enhance images includes: Histogram adjustment, Fourier transforms, Brightness normalization, and Homomorphic image filtering.

**2.4.1 Histogram Adjustment** A gray-level histogram of an image is a graphical function that describes the frequency of occurrence of each gray level pixel in the image [5:70]. Each pixel is quantized from 0 to  $n$ , where  $n$  depends on the number of bits used to describe the pixel. The histogram can adjust the pixel levels in the images and transform the image by equalizing the distribution of the gray levels. Figure 2.2 shows an example of a histogram.



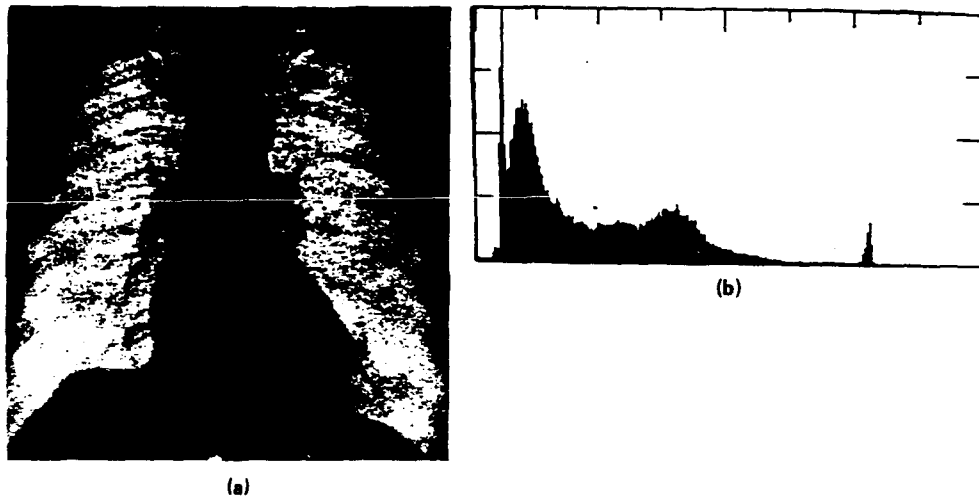


Figure 2.2. Example of Image and Corresponding Histogram [5:70].

**2.4.2 Fourier Transforms** Most vision system's properties are easier to analyze in a frequency domain than in a spatial domain. The Fourier transform is commonly used to transform information from the spatial domain  $(x, y)$  to a frequency domain  $(\omega_x, \omega_y)$ . The two dimensional continuous Fourier transform is

$$F(\omega_x, \omega_y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{j(\omega_x x + \omega_y y)} dx dy \quad (2.13)$$

where  $\omega_x$  and  $\omega_y$  are angular spatial frequencies [6:399]. The Discrete Fourier transform (DFT) for a square array of  $N \times N$  image elements is

$$F(\omega_x, \omega_y) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \frac{(\omega_x x + \omega_y y)}{N}} \quad (2.14)$$

with the inverse DFT expressed as

$$f(x, y) = \frac{1}{N} \sum_{\omega_x=0}^{N-1} \sum_{\omega_y=0}^{N-1} F(\omega_x, \omega_y) e^{-j2\pi \frac{(\omega_x x + \omega_y y)}{N}} \quad (2.15)$$

[17:601-602]. Fourier transforms can be used to [19:95]:

- enhance edges (high pass filters),
- reconstruct images from projections, and
- aid object recognition based on spatial frequency content.

High frequency components of a Fourier transform contain the information on quickly varying detail in an image [5:25]. Therefore the edges in an image can be enhanced by emphasizing the high frequency components of the image. Fourier transformations can be accomplished easily by optical lenses as the image is acquired. Digital implementation of the Fourier transformation function is slower and currently may not be suitable for real time object recognition.

*2.4.3 Brightness Normalization* Brightness Normalization is the process of adjusting the brightness of image pixels based on the local or global average of those pixels. Global thresholding is one method to solve the problem of variations in lighting. Global thresholding determines the median value of all the pixels in the image and then adjusts each pixel magnitude up or down based on the median value of the image. [11:282-283]. Global thresholding adjusts the whole image and may result in unnecessary processing if only a small portion of the image is in shadow, and needs normalization.

To address that issue Lambert developed a different method to normalize the brightness of an image, called local thresholding or Lambertization [12:3-17]. His algorithm calculates the average brightness in a square-shaped neighborhood around a pixel and resets the pixel value by the following equation.

$$pixelvalue = 128 + (pixelvalue - neighborhoodaverage) \quad (2.16)$$

This algorithm allows the brightness adjustment of small portions of an image and performed quite well in adjusting the threshold of the image used to perform face recognition [12:17]. By adjusting the image thresholds, either globally or by Lambertization, brightness normalization can remove false edges and improve edge detection in shadows. Figure 2.3 shows an example of a brightness scaled image.

2.4.4 *Homomorphic Image Filtering* [7:84-86] Homomorphic image filter is based on the fact that an image  $f(x, y)$  can be modeled as a product of a reflectance  $a(x, y)$  and an illumination  $b(x, y)$  [7:84-86].

$$f(x, y) = a(x, y)b(x, y) \quad (2.17)$$

The filtering steps include taking the logarithm and Fourier transforms of an image, running the transform through a filter, and then performing the inverse transforms. Taking the logarithm of the incoming image adjusts the scaling of the image. The Fourier transform is used to enhance the edges in the image by identifying the frequency relationships. Figure 2.4 shows the processing steps. Homomorphic filtering can reduce the effect of differing lighting conditions and improve the contrast of an image. Figure 2.5 shows an example on a filtered image. By taking the logarithm of both sides of Equation 2.18, the

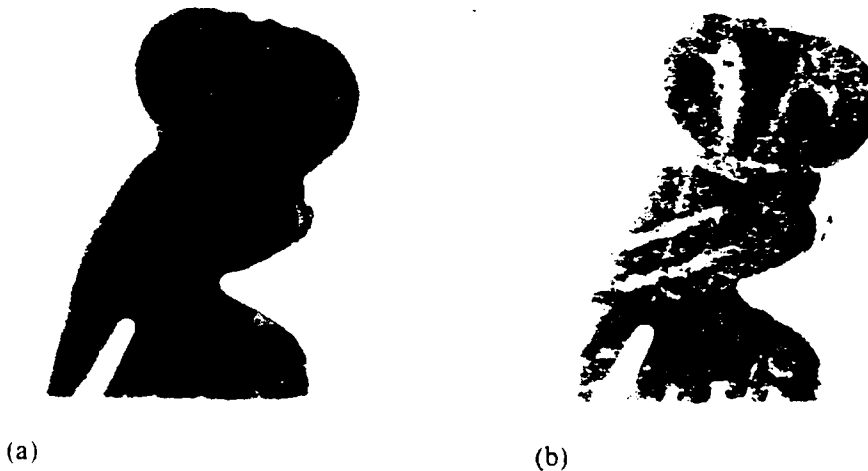


Figure 2.3. Example of a Brightness Normalized Image. a) Original b) Normalized [7:64].

multiplication is converted to addition.

$$\ln f(x, y) = f'(x, y) \xrightarrow{\mathcal{F}} A'(\omega_x, \omega_y) + B'(\omega_x, \omega_y) \quad (2.18)$$

Taking the Fourier transform of  $f'(x, y)$  takes advantage of the linearity property of the

Fourier transform.

Homomorphic image filtering provides an excellent way of improving the image contrast by highlighting edges in the image. Homomorphic filtering is unnecessary and may slow down processing when other edge detection and recognition processes have to be accomplished to identify the object of interest.

## 2.5 *Object Recognition Implementations*

AFIT has researched numerous pattern recognition algorithms to address the problem of object recognition. Three different recognition implementations, AFIT Distortion Invariant Pattern Recognition Algorithm, Autonomous Face Recognition Machine, and the Robotic Visual Servoing System were reviewed to determine the best approach to use for this thesis effort.

*2.5.1 AFIT Distortion Invariant Algorithm* The AFIT Distortion Invariant Pattern Recognition Algorithm (DIPRA) (see Figure 2.6) has demonstrated the ability to handle position, rotation, and scale problems in target recognition [23:63-65]. Step one of the algorithm computes the two dimensional Fourier transform to form a position invariant feature space. The second step performs a rectangular to polar coordinate conversion to handle differences in rotation between the target and the template. The third step performs logarithmic scaling to handle scale differences. These first three steps of the AFIT algorithm form the position, rotation, and scale invariant feature space which is used in step four which performs a correlation of the template with the input image to find the targets of interest. The last four steps of the algorithm are used to reconstruct the original image with the target enhanced. The implementation is similar to homomorphic image filtering. Both techniques use logarithmic and Fourier transformations in their image processing, but the AFIT algorithm also adds a rectangular to polar coordinate transformation. Both algorithms produce an enhanced image, with the AFIT algorithm adding the capability to do object recognition. The AFIT DIPRA algorithm has shown promise for target recognition applications but is very time consuming to implement and consequently would not work well in a real-time pattern recognition environment.

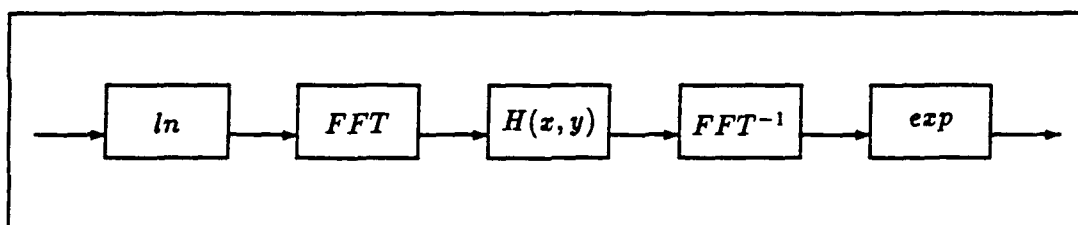


Figure 2.4. Processing Steps in Homomorphic Image Filtering [7:86].

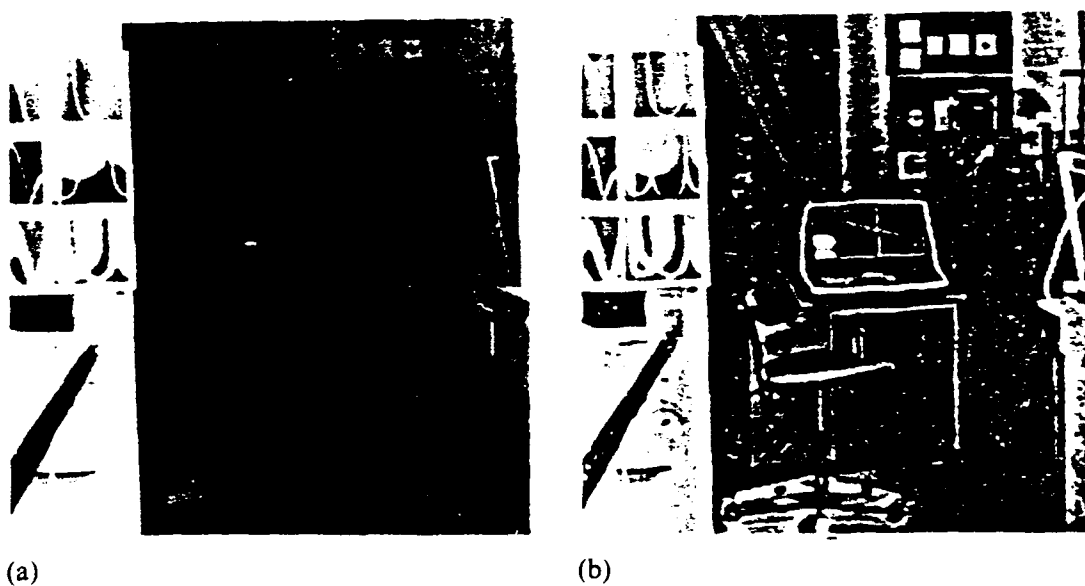


Figure 2.5. Example Homomorphic Filtering a) Original b) Filtered [7:86].

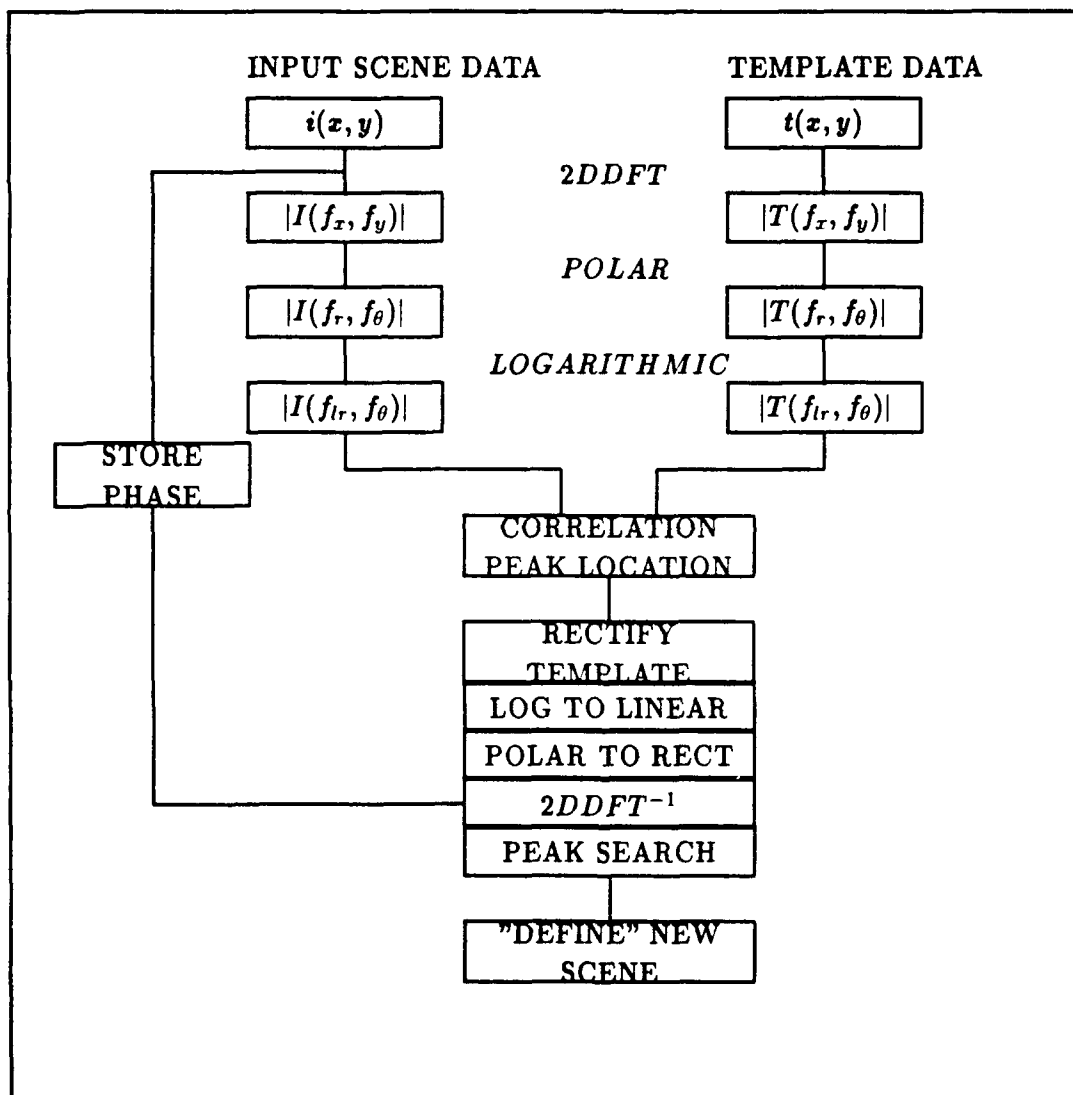


Figure 2.6. The AFIT Distortion Invariant Pattern Recognition Algorithm [23:64].

**2.5.2 Autonomous Face Recognition Machine (AFRM)** The Autonomous Face Recognition Machine (AFRM) uses Fourier transforms to build a reliable feature set for recognition of faces [20:9]. The AFRM implements a five step process for face recognition. Step one acquires the facial image. Step two locates the face by looking for the features that make up a face: two eyes, a nose, and a mouth, and then performs a brightness normalization. The third step preprocesses the image by performing contrast enhancement and scaling of the face. The fourth step makes six face windows of the face to be recognized. Step five performs a two dimensional discrete Fourier transform of the face to compare against the data base for recognition. The best recognition percentage achieved with the AFRM was 87%, which is unacceptable for most applications. Accuracy of near 100% is needed to make any recognition process acceptable. The AFRM is a time-consuming process because of the large data base that must be maintained and searched. As the data base grows, with more faces being added, the process speed is further reduced. If only a small data base is maintained and an efficient data base search algorithm is developed, then the AFRM algorithm could be extended to other recognition processes.

**2.5.3 Robotic Visual Servoing System (RVSS)** The RVSS used white indicators placed on three sides of the refueling port (see Figure 2.7) to aid in refueling port recognition [22:3.1-3.13]. The indicator approach was chosen to decrease the complexity of the recognition process and reduce the processing time. The first step, after the image was obtained, was to perform a histogram on the image and store the data in an array. Pixels above a predetermined threshold were classified as white and everything else was considered black, see Figure 2.8. The next step was to locate the center of mass of all the white pixels. Once the center of mass was determined, the algorithm looked for the indicator above the center of mass. If the algorithm found the first indicator, it would look for the second indicator to the left of the center of mass. If the first two indicators were found, the algorithm would look for the last indicator to the right of the center of mass. The length of the top indicator was used to determine the range to the refueling port. If all three indicators were not visible in the image, the refueling port was considered not present. If the port was not located, the RVSS would move the robot arm to a new location, in its search pattern, stop and look again for the port. The RVSS algorithm was very susceptible



Figure 2.7. Refueling Port with Indicators [22:3.8].

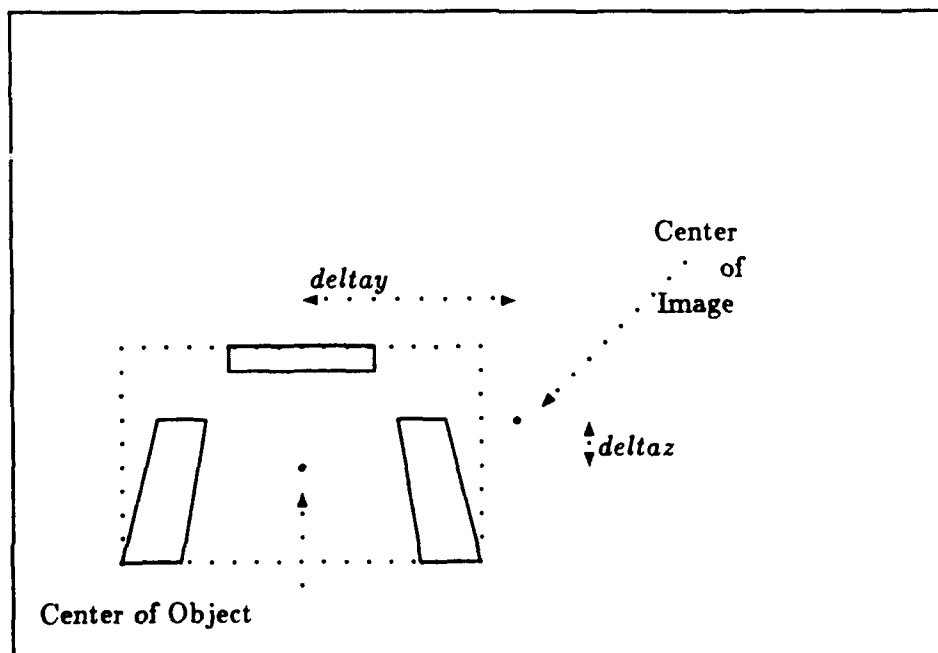


Figure 2.8. Relationship of Indicator Parameters [22:3.12].



to lighting condition. Insufficient light or shadows resulted in the refueling port not being recognized even though the port was clearly visible in the image. If the RVSS found the port, it computed an offset from the center of the scene and placed the refueling arm into position. Once the robot arm was moved into position, the recognition process stopped and if the port moved the RVSS would not notice the movement and adjust the arm position. Currently, there is no capability to track a moving refueling port. The RVSS algorithm requires approximately five seconds to recognize the refueling port under ideal lighting. The most time consuming portion of the RVSS is the input/output communication processing between the ITEX board and the Vax computer.

*2.5.4 Fuel Interchange System[13:159-174]* Abidi and Gonzalez worked on a Fuel Interchange System (FIS) to autonomously conduct refueling operations for spacecraft [13:159-174]. The refueling receptacle holds four light guiding targets used to determine the position and orientation of the port (see Figure 2.9). For the FIS system to determine

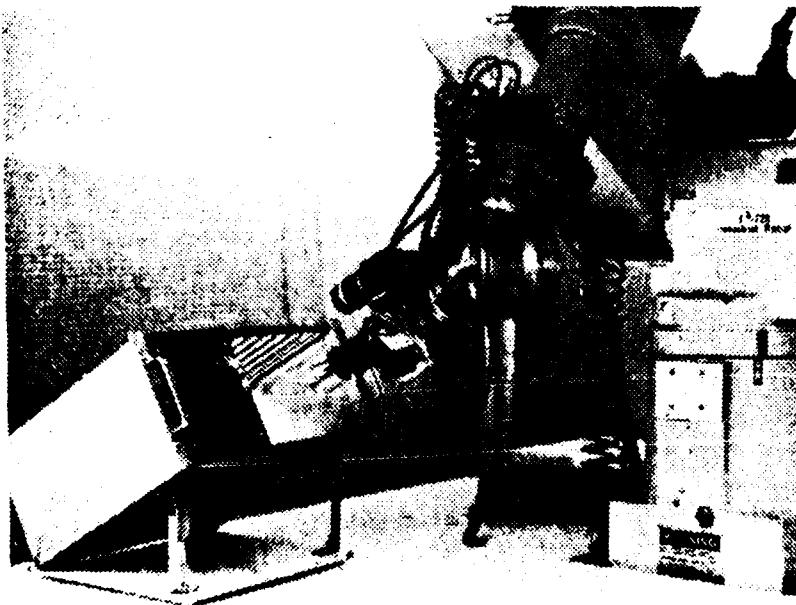


Figure 2.9. Refueling Receptacle with Point Targets [13:170].

the location of the fuel receptacle, the receptacle has to be positioned in the field of view of the camera. A human operator moves the robot so that the receptacle is visible, then the robot is put into autonomous mode to refuel the system. The concept of light point targets around the receptacle is similar to the passive rectangular indicators used by Shipman's RVSS system. The active admitting lights around the receptacle would not be acceptable for robotic aircraft refueling applications. The range information is also determined similarly by the distance between light points for the FIS vs length of the top indicator for the RVSS.

## *2.6 Summary*

This review highlighted several image processing and object recognition techniques. Several different implementations were presented to give examples of previous accomplishments in the area of object recognition. Image thresholding, brightness normalization, scaling, edge detection, and template matching all may play important parts in the recognition of the refueling port. The use of these concepts can reduce the affect of lighting and are vital for successful implementation of a light invariant recognition system for robotic aircraft refueling.

### *III. Refueling Port Recognition*

#### *3.1 Introduction*

An important process in a robotic aircraft refueling system is the recognition of the refueling port. All other aspects of the refueling system build on knowing where the refueling port is located. To aid in the identification and location of the refueling port, rectangular indicators (see Figure 3.1) were placed around the simulated refueling port [22]. Shipman's previous research on recognizing the indicators required the port to be in ideal lighting. If the port was in shadows or there was too much light, the recognition process would fail. Ideal lighting that minimizes shadows can't be guaranteed, so a brightness invariant recognition algorithm needs to be developed. Searching for intensity changes in an image is one way of handling the variable brightness problem. Since edge detection relies on texture or intensity changes between neighboring regions to identify edges, the edges should be recognizable in a wide range of lighting conditions. This research effort investigated the uses of edge detection to make a brightness invariant port recognition system (BIPRS). Also, in order to develop an efficient search strategy, the recognition algorithm must be able to recognize the refueling port when it is partially visible in the image. This chapter presents the theory for the recognition process used to locate and identify the refueling port and is divided into the following sections:

- *Initial Processing* details the acquisition of images and how the pixel information is transformed into edges.
- *Edge Processing* details how edges are concatenated together and how the closed loops are formed to find the indicators.
- *Loop Determination* describes how geometric approximation is accomplished to find the shape of the closed loops.
- *Port Recognition* describes how the closed loop are analyzed to determine if the refueling port is present.
- *Refueling Port Search* describes how the search for the refueling port is accomplished.

The port recognition process described in this chapter will provide a brightness invariant port recognition capability that is vital to demonstrate a viable robotic aircraft refueling concept. The software logic for the theory presented in this chapter is in appendix A and the C code is in an internal report generated as a result of this research [18]

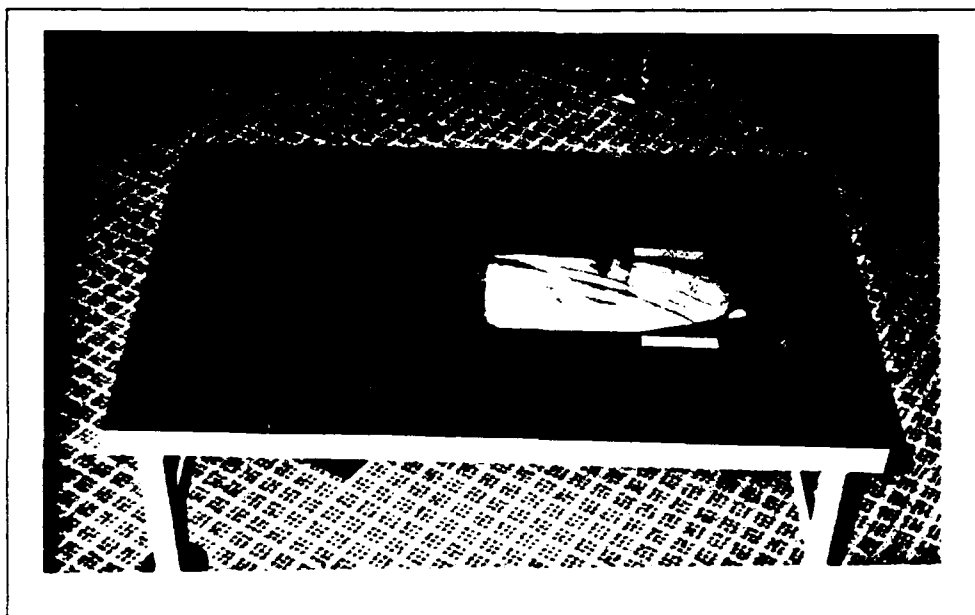


Figure 3.1. Refueling Port with Rectangular Indicators.

### 3.2 Initial Processing

The initial processing step in the recognition process consists of the acquisition of the image, Sobel edge detection, and Non-maximal suppression to locate potential edge pixels. The *Sobelmain* routine controls the initial processing step of BIPRS. *Sobelmain* first calls the *Sobeltest3* subroutine to initialize the ITEX board then commands the ITEX board and to obtain an image.

**3.2.1 Image Acquisition** All vision processing for this research was performed using an Image Technology Series 100 image processing board hosted in a VAXstation III computer. The ITEX image processing board is used to acquire the image. The image we

digitized into a two-dimensional array (512 x 512 pixels) and placed in a frame buffer memory for further processing [9]. Location (0,0) represents the upper left corner of the image and location (512,512) represents the lower right corner of the image. A standard monitor can only display the top 480 horizontal lines of the 512 by 512 array (see Figure 3.2). Since only the top 480 lines are visible, and for ease of analysis of system performance, these lines are the only ones used for this research effort. The frame buffer stores a 12 bit representation of each pixel but we use only the first 8 bits. The upper four bits are used for special ITEX functions.

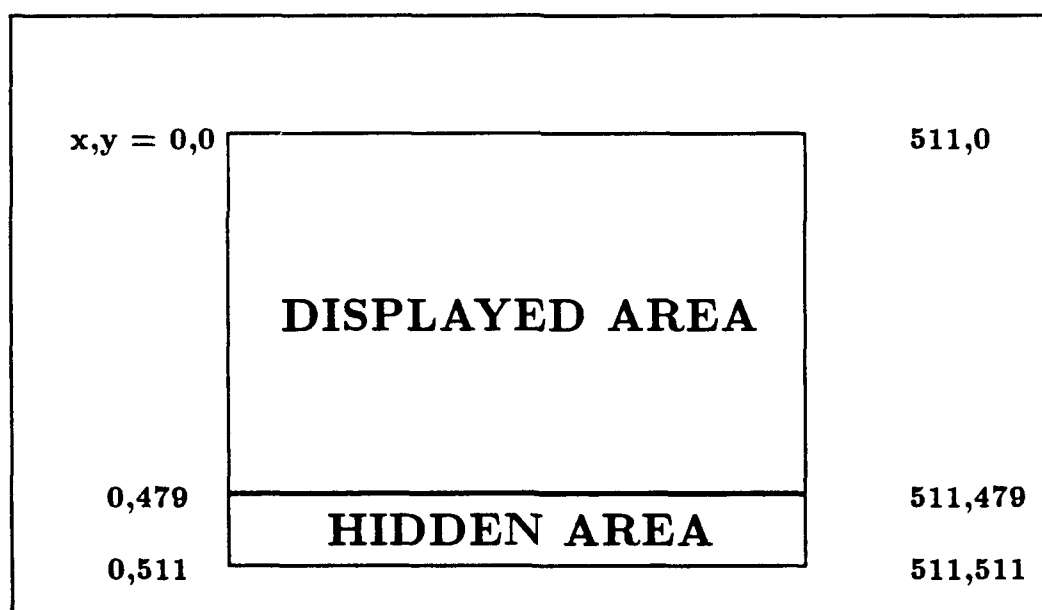


Figure 3.2. Video Coordinate System [9:1-6].

After the image is acquired, the *Sobelmain* routine commands the ITEX board to perform Sobel edge detection and stores the edge values in a 512 X 480 array. The ITEX board has two edge operator functions available to perform edge detection, a Roberts Edge Filter and a Sobel Edge Operator. The Sobel edge operator (SEO) function of the board

was chosen to obtain an edge representation of the image because it provides a finer, more continuous edge than the Roberts operator [9].

**3.2.2 Sobel Edge Detection** The Sobel edge detection provides the edge representation in an image by sliding a pair of three by three masks across the image (see Section 2.2). Figure 3.3 shows an example of an edge in an image. The edge is identified by a change in intensity of the pixels in the image. When the SEOs  $D_x$  and  $D_y$  are convolved with the image, they produce the relative strengths in the  $x$  and  $y$  directions for each pixel.

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$D_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.2)$$

The Sobel edge in Figure 3.4 is formed from the  $x$  and  $y$  strengths. For pixel location (3,11) the strength in the  $x$  direction is 7 and the strength in the  $y$  direction is -13. Using Equation 2.10, the pixel strength would be  $(7^2 + 13^2)^{1/2} = 14$ . The direction of the edge is given by  $\alpha = \tan^{-1}(-13/7)$  and equals -61 degrees. Figure 3.5 shows how the direction value equates to the direction of the edge. An edge direction of -61 degrees would equate to a Sobel edge direction of 6 from the SEO. The SEO function of the ITEX board gives a strength and direction for each edge pixel identified and uses only the first 8 bits of each pixels value (see Figure 3.6). The strength ranges from zero to thirty-one and is stored in bits three to eight. The direction ranges from zero to seven and is stored in bits zero to two. The Sobel edge values are read into a 512 X 480 array for further edge processing. After Sobel edge detection, *Sobelmain* calls *NMSmain* to perform Non-maximal suppression of the Sobel edges.

**3.2.3 Non-maximal Suppression** Non-maximal suppression is widely accepted method for thinning edges. The edge formed by the Sobel edge operator is between three to five

	00	01	02	03	04	05	06	07	08	09	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	3	7
4	0	0	0	0	0	0	0	0	3	7	10	10
5	0	0	0	0	0	0	3	7	10	10	10	10
6	0	0	0	0	3	7	10	10	10	10	10	10
7	0	0	3	7	10	10	10	10	10	10	10	10
8	3	7	10	10	10	10	10	10	10	10	10	10
9	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10
11	10	10	10	10	10	10	10	10	10	10	10	10

Figure 3.3. Example of an Image Edge.

pixels wide. The pixel strength across the edge increases to a maximum value then decreases (see Figure 3.7). When the edge is more than one pixel wide, it is difficult to tell where the actual edge is for further edge processing. A process called non-maximal suppression (NMS) is used to thin the edge to one pixel. Figure 3.8 shows the Sobel edge after non-maximal suppression. The NMS algorithm, *NMSmain*, compares a candidate edge pixel against its two neighbors. The neighboring pixels are determined based on the direction of the edge given by the SEO. If the candidate pixel's magnitude is greater than both its neighbors or equal to its left neighbor the pixel is added to the MARK array for additional processing. Pixels which don't pass the test are ignored. This process is repeated for every positive pixel in the pixel array. Stray pixels, called noise, are prevalent at some lighting conditions. This noise can interfere with edge processing and prevent recognition of objects in the image. The image is thresholded, at the same time as the NMS is accomplished, to remove noise from the image. All the pixels that are below the threshold value of three are removed from further processing. There is nothing magical about the threshold value of three. The threshold value was determined by experimental

	00	01	02	03	04	05	06	07	08	09	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	4	14	0
3	0	0	0	0	0	0	0	4	14	29	40	0
4	0	0	0	0	0	4	14	29	40	40	29	0
5	0	0	0	4	14	29	40	40	29	14	4	0
6	0	4	14	29	40	40	29	14	4	0	0	0
7	0	29	40	40	29	14	4	0	0	0	0	0
8	0	40	29	14	4	0	0	0	0	0	0	0
9	0	14	4	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.4. Example of a Sobel Operator Edge.

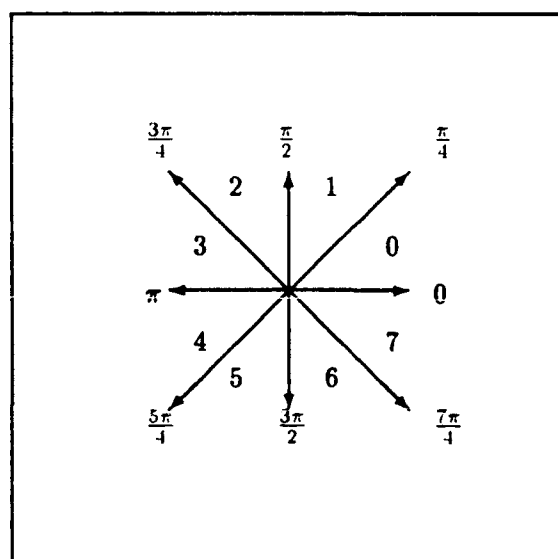


Figure 3.5. Sobel Edge Operator Directions.



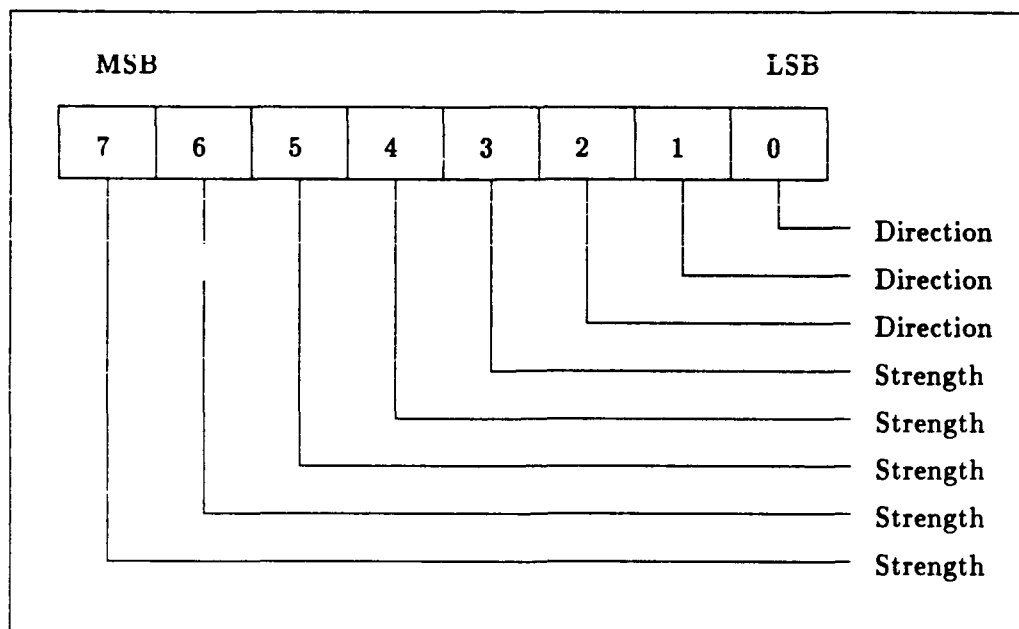


Figure 3.6. Strength and Direction Bit Storage [9:9-32].

evaluation of different threshold values over the range of brightness variation that can be expected in the system.

**3.2.4 Edge Linked-List Formation [4]** After NMS, *Sobelmain* calls *Linkmain* to form a linked-list of edge pixels. All the pixels that are left after the NMS process are searched and combined into a linked-list of pixels that form small edges. First, the MARK pixel array from the NMS process is searched for positive pixels. Once a positive pixel is found, it forms the head of a linked-list of connected pixels. The neighbors of the first pixel are searched for additional positive pixels. If only one positive pixel is found in the surrounding area, this pixel is linked to the previous pixel and then its neighbors are searched for pixels. As each pixel is added to a linked-list of connected pixels the pixel's position in the array is set to zero. This process is repeated until the last pixel in the linked-list has no positive neighbors. The pixel with no positive pixel neighbors forms the tail of the linked-list of

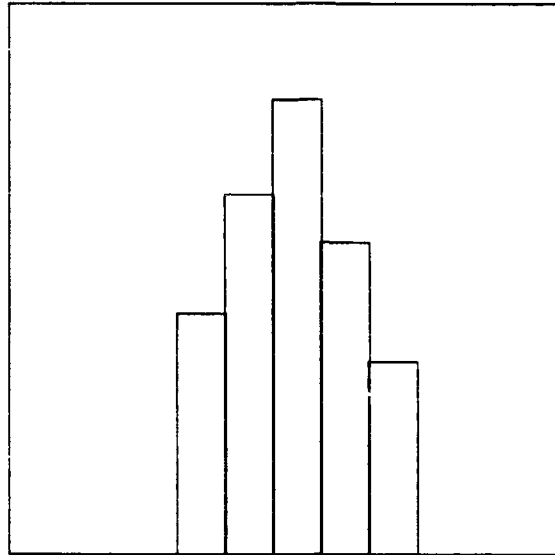


Figure 3.7. Example of Sobel Edge Strength Distribution.

	00	01	02	03	04	05	06	07	08	09	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	40	0
4	0	0	0	0	0	0	0	0	40	40	0	0
5	0	0	0	0	0	0	40	40	0	0	0	0
6	0	0	0	0	40	40	0	0	0	0	0	0
7	0	0	40	40	0	0	0	0	0	0	0	0
8	0	40	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.8. Non-Maximum Supression Thinned Sobel Edge.

pixels. Once a tail of a linked-list of pixels is found, the linked-list is added to a linked-list of small edges for additional processing. Then the array is searched for another positive pixel to form another linked-list of pixels and the procedure is repeated until no positive pixels remain in the original pixel array. Once all the pixels are combined into small edges, the linked-list of edges is sorted to place the larger edges on top and the smaller edges on the bottom.

### 3.3 Edge Processing

In the edge processing stage the list of small edges is searched for edges that can be combined together by replacing missing pixels to form closed loops. This process is accomplished in the *Comblist* subroutine called from *Linkmain* routine.

**3.3.1 Line Merging** Line merging combines edges that have end points in close proximity to each other. The linked-list of edges, formed in the edge formation stage, is searched for edges that can be combined. The end of the first edge in the linked list is compared with the ends of other edges. If there is only one edge that has an end within close proximity of the end of the target edge, the two edges are combined (see Figure 3.9). If more than one connection is possible, no combination is accomplished. The process is repeated until all possible connections are made. Edges can be combined for gaps between edge segments as large as seven pixels.

**3.3.2 Closed Loop Formation** Closed loops are formed in the *Comblist* subroutine. Closed loop formation attempts to form closed loops from edges with a head and tail in close proximity to each other. If the head of the linked-list is within five pixels of the tail of the list, then the loop is completed and added to a linked-list of closed loops. If both ends of edge are on the border of the image the loop is closed by inserting pixels until the two ends are connected and then the loop is added to the closed loop list. The formation of closed loops is important in recognizing the rectangular indicators. Once all the edges are combined in the edge concatenation stage, the linked-list of edges is searched for possible closed loops that could represent one of the indicators used for recognizing the refueling

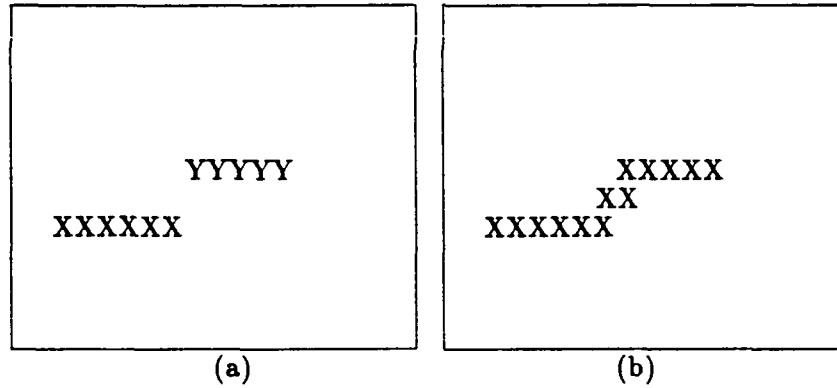


Figure 3.9. Example of Edge Combination. (a) Before Combination. (b) After Combination.

port [3]. After the closed loops are formed, *Linkmain* calls *Seachmain* routine to determine the shape of closed loops and search for the refueling port.

### 3.4 Loop Identification

The closed loops formed from the previous step are searched to find the rectangular indicators. This is accomplished in the *Lstsrch* subroutine. The centroid of the closed loops is calculated by summing the  $x$  and  $y$  position of all the pixels in the loop and dividing by the number of pixels in the loop.

$$C_{mass_x} = \frac{\sum_{i=1}^N x_i}{N} \quad (3.3)$$

$$C_{mass_y} = \frac{\sum_{i=1}^N y_i}{N} \quad (3.4)$$

After the centroid is calculated, the shape of the closed loop is determined by arbitrarily dividing the closed loop into two separate segments, then searching each segment for the point on the loop that is the maximum distance from the line connecting the end points of the two segments. For example, in Figure 3.10, a line is first drawn from point 1 to point 2.

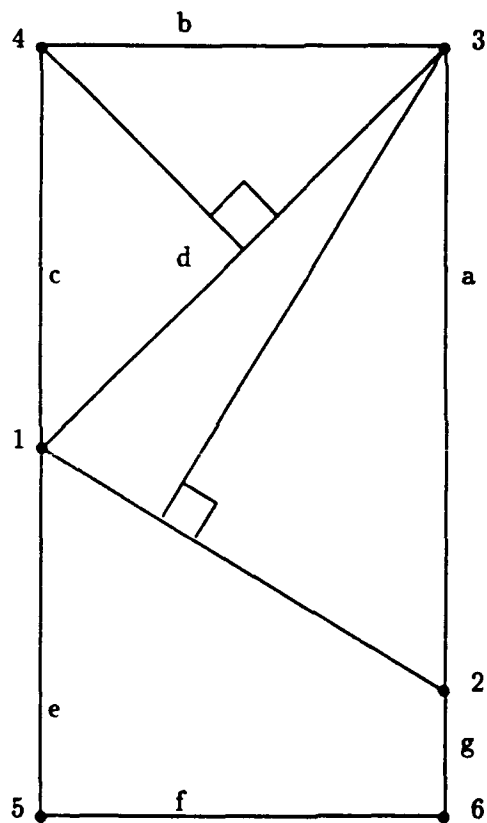


Figure 3.10. Shape Approximation Process for Closed Loops.

Each pixel on the top half of the closed loop, is searched to find the pixel that is farthest away from the line connecting the first two points. Once point 3 is found, lines are drawn from the previous two points ( 1 and 2) to point 3. The process of finding the maximum point from the lines is repeated for the next segment of the closed loop. This process is repeated recursively until the maximum distance between the line connecting the previous two points and the pixels on the closed loop is less than two pixels. Figure 3.10 shows that line segments a,c,d,e,f, and g would approximate the rectangular indicators that are being searched for. The approximation of the shape of the closed loop is independent of where the closed loop is segmented to get the approximation. Once all the line segments that approximate the closed loop are determined, the slopes of the lines are compared with its neighbor's slope and if the arctan of the slopes are within 5 degrees of each other then the two line segments are combined to form one line. In Figure 3.10, line segments c - e and a - g could be combined thus representing the closed loop of interest by four line segments. The process is repeated for each closed loop identified in the image.

### 3.5 Knowledge Driven Port Recognition

*Searchmain* calls *Findport* to search for the attributes of the port indicators. The line approximation of closed loops are first searched to find the top indicator ( *Findtop* subroutine). The top indicator is searched for first because it is the most important for determining the range to the refueling port. The attributes or constraints of the top indicator are:

- The closed loop is rectangular (number of lines approximating the loop equal to four with two parallel long edges).
- The rectangle has a ratio of length:width of approximately 3:1.
- The slope of the long edge is greater than 1.0.

A loop is identified as the top indicator if all the above attributes are present. There is nothing magical about the three to one ratio of the length to width. The length of the top indicator was made equal to the width of the top edge of the refueling port and the width

of the indicator is equal to the width of gray tape used for the indicators. After the top indicator is searched for, the closed loops are searched for the two side indicators ( *Findnzi* subroutine). The side indicators have the following attributes or constraints:

- The closed loop is rectangular (number of lines approximating the loop equal to four with two parallel long edges).
- The rectangle has a ratio of length:width of approximately 4:1.
- The slope of the long edge is less than 1.0.
- (if Top indicator found) The slope of the long edge of the indicator is perpendicular to the slope of long edge of the top indicator.

The ratio of the length to width of the side indicators was made different from the top indicator to provide one additional unique attribute. If the slope of the line connecting the center of mass of the side indicator to the center of mass of the top indicator is positive, then the side indicator is identified as the left indicator, otherwise the indicator is identified as the right indicator. If the top indicator was not located first, there is no way to determine whether the side indicator identified is the right or left indicator. If all three indicators are found, the center of mass of all the indicators is calculated. The *deltay* and *deltaz* from the center of the image to the center of mass of the three indicators is calculated to identify the change in robot position required to center the indicators (see Figure 3.11). Range is determined based on the length of the top indicator. All three values, *deltay*, *deltaz*, and *range* are used to determine the required position change [22].

### *3.6 Searching for the Port*

The search for the refueling port is accomplished autonomously using the pattern in Figure 3.12. The search pattern assumes that the robot was initially located close to the port. Initial placement could be by a human operator or by global autonomous positioning system. The robot continues to search for the port until the BIPRS identifies the port or the search is terminated manually. If the robot reaches one of its mechanical stops during the search, the direction of the search is reversed. If during the search for the port, at

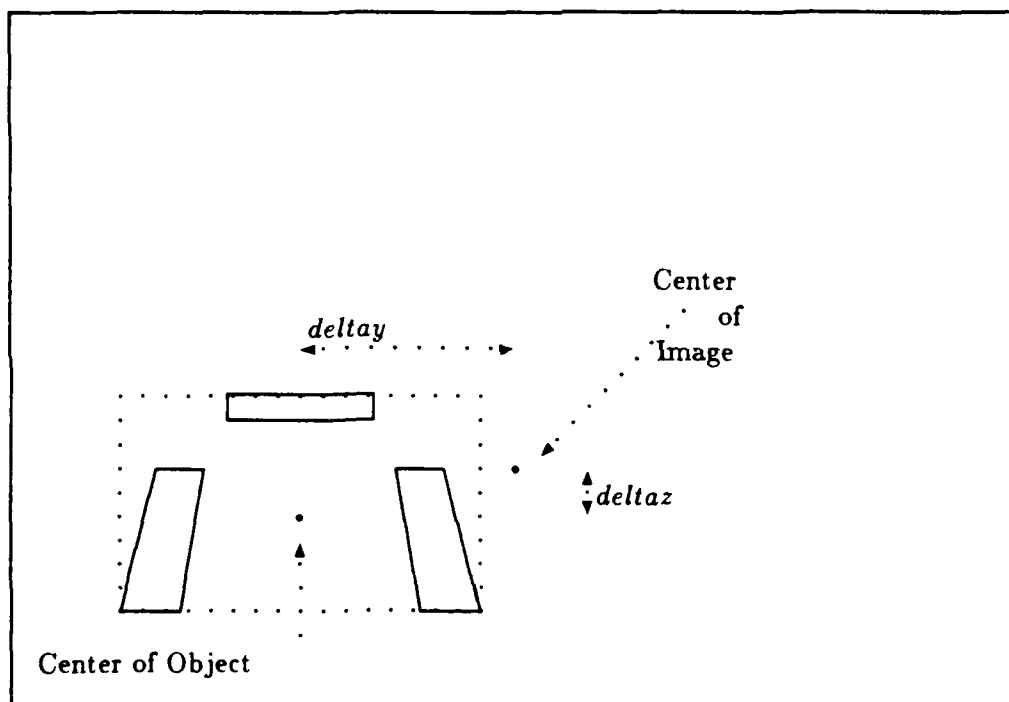


Figure 3.11. Indicator Parameter Relationship [22].



least two possible indicators are identified the robot will deviate from the search pattern to check for the port. If the port is not found the robot will resume the search from the point where it last searched for the port. If two indicators , or one indicator with a part

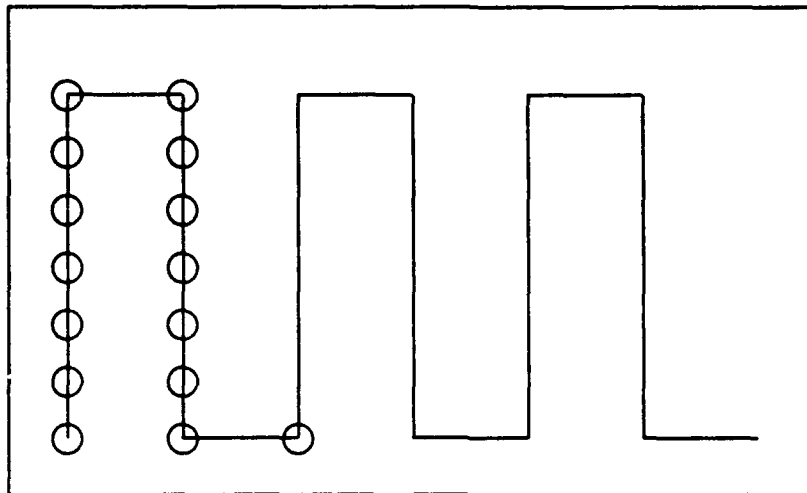


Figure 3.12. Search Pattern [22].

of a possible second indicator are identified, the center of mass is calculated and the robot arm position is adjusted to center that point in the image. Once the image is centered the BIPRS takes a second look to verify that the refueling port is present before moving the arm to the refueling port or continuing the search for the port. Figures 3.13-3.15 show the process of homing in on the refueling port. In Figure 3.13 only two indicators are visible. The vision system departs from the search pattern and moves the two indicators towards the center of the image. Figure 3.14 shows that a third indicator is now partially visible and the BIPRS indicates that there are three indicators, but since one is partially occluded, the system again centers the indicators. Figure 3.15 now shows that all the indicators are now visible and the port is recognized. Figures 3.16 and 3.17 show two other conditions that would result in the recognition process departing from the search pattern to center the visible indicators in the image.

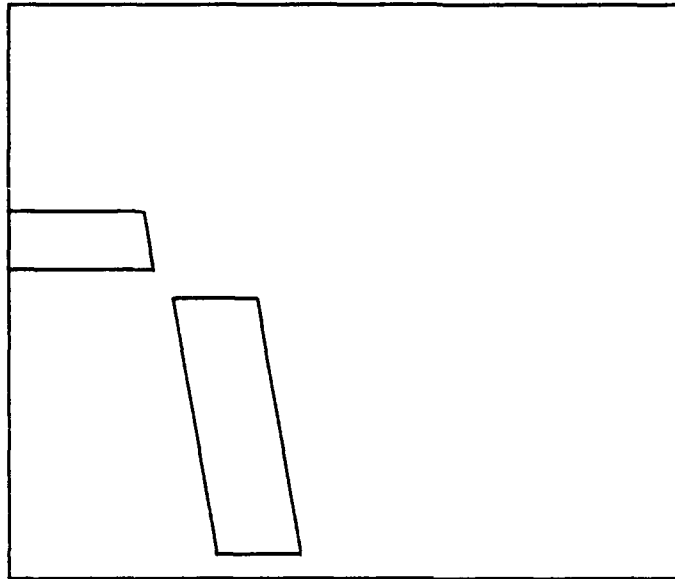


Figure 3.13. Image With Only Two Indicators Visible.

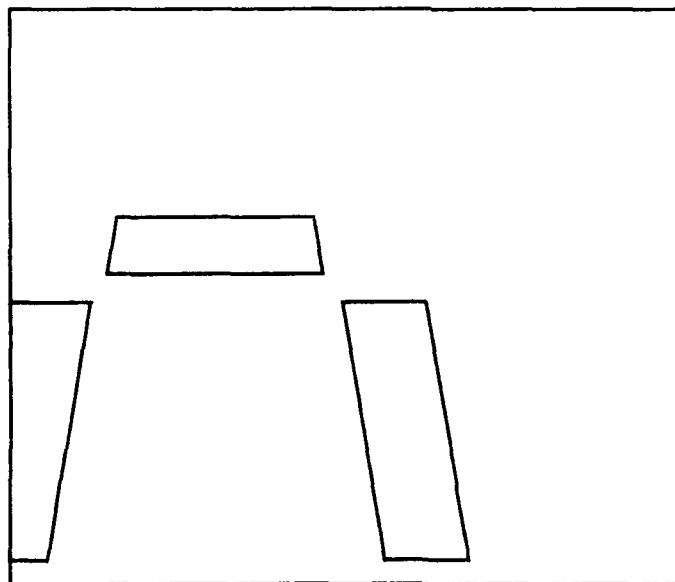


Figure 3.14. Image With Two Indicators And a Partially Visible Third Indicator.

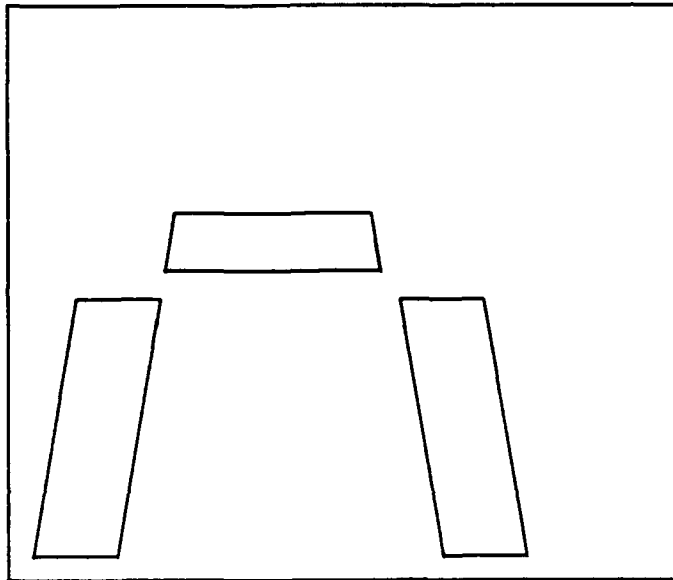


Figure 3.15. Image With All Three Indicators Visible.

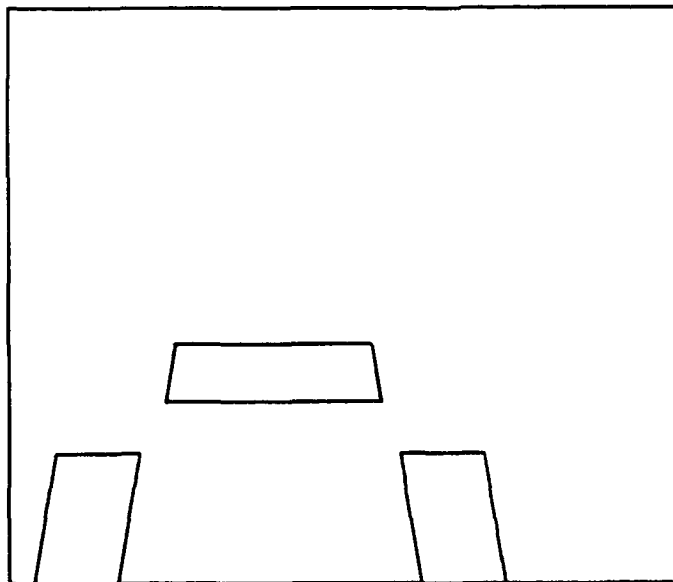


Figure 3.16. Image With Top And Two Partial Indicators Visible.

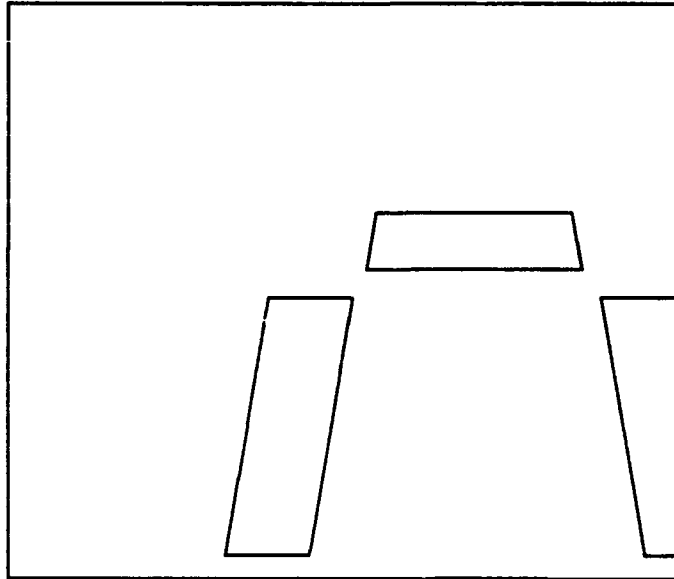


Figure 3.17. Image With Two Indicators And a Partially Visible Third Indicator.

### 3.7 Summary

This chapter described the acquisition of the vision information, detailed the implementation of the port recognition algorithm used to identify the refueling port, and outlined the search strategy used to locate the port. The results of implementing the port recognition algorithm presented in this chapter are highlighted in Chapter 4.

## *IV. Results and Discussion*

### *4.1 Introduction*

The primary goal of this thesis effort was to develop and demonstrate a brightness, invariant port recognition system (BIPRS) to locate and identify an aerial refueling port under a wide range of lighting conditions. In addition to brightness invariance, the BIPRS was developed to be size and rotation invariant, i.e. to recognize the refueling port regardless of its position. The task was accomplished by using edge detection techniques, current vision technology and the RVSS's trajectory planning software. The BIPRS development was divided into three processing stages:

- image acquisition and initial processing,
- edge processing, and
- port recognition.

Each stage of the recognition process was developed and tested to ensure an effective port recognition system. The results of those tests are divided into the following critical areas:

- how the edge detection improved the recognition of the refueling port under a wide range of lighting conditions,
- how thresholding reduced the effect of noise on edge processing,
- how edge detection helped identify the port when the port was rotated away from the robot,
- how the recognition of the port was not dependent on the size of the port,
- how the port was recognized when indicators are only partially visible, and
- how the search strategy was used to locate the refueling port.

These issues are addressed sequentially in the remainder of this chapter. Since the primary goal of this research effort is the concept demonstration of brightness invariant recognition, processing speed was not considered to be critical. The time numbers presented in this

chapter are intended to represent percent of computation time for each function which will not change significantly without parallel processing.

#### *4.2 Initial Processing*

The initial processing stage of the BIPRS is the most important part of the recognition process. This stage is responsible for the acquisition of the image to be processed, commanding Sobel edge detection, performing non-maximal suppression to thin edges, and thresholding the image. Figure 4.1 shows an example of the initial image acquired by the ITEX image processor and Figure 4.2 gives an example of the image after Sobel edge detection. This initial processing stage is the most time consuming stage in the recognition process taking 50 out of the 90 seconds of total processing time. The acquisition of the image and performing the Sobel edge detection requires approximately 40 seconds. Performing NMS and thresholding uses the other 10 seconds.

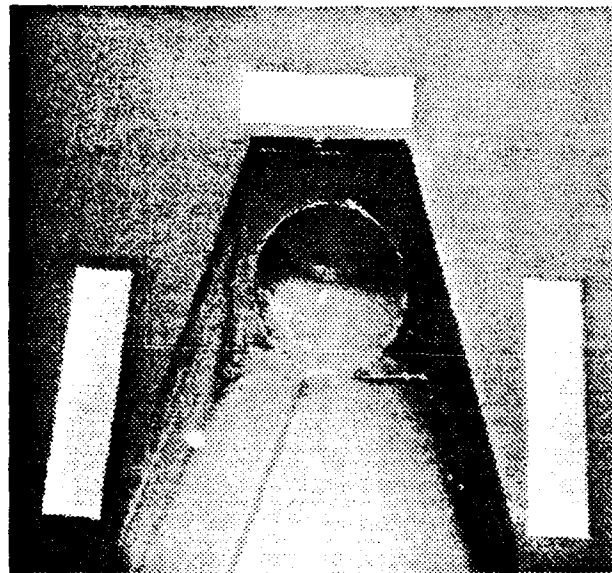


Figure 4.1. Original Image of the Refueling Port.

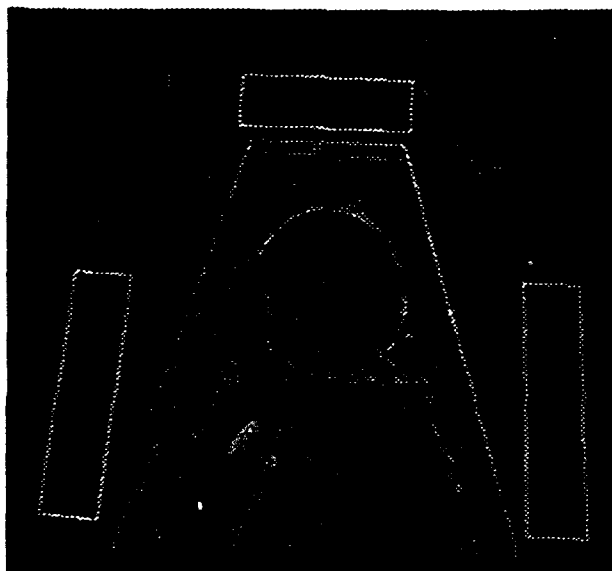


Figure 4.2. Example of Sobel Edge Image.

**4.2.1 Non-maximal Suppression** Non-maximal suppression takes the Sobel edges from the previous step and thins the edges to one pixel. NMS is required because it is difficult to process edges when they are wider than one pixel wide. NMS takes 10 seconds to complete because it has to search a 512 x 480 array pixel values and compare each pixel to its neighbors to find the maximum pixel of the edge. A more detailed description of NMS is contained in Chapter 3.

**4.2.2 Thresholding** Thresholding eliminates noise in the images and helps isolate the indicators from the rest of the image. Figure 4.3 gives an example of the non-maximum suppression image in normal lighting before thresholding. The image in Figure 4.3 shows that it is important to threshold an image to remove noise. Without thresholding, it is hard to identify the rectangular indicators from all the stray pixels in the image, but with thresholding the indicators are easy to see. Figure 4.4 shows the same lighting condition with thresholding. The strength of the pixels in the image could vary from zero to thirty-one. Thresholding removes any pixel that is below the selected value from further processing. The threshold value of three was set based on experimental runs. The



Figure 4.3. Non-Maximal Suppression Image Without Thresholding.

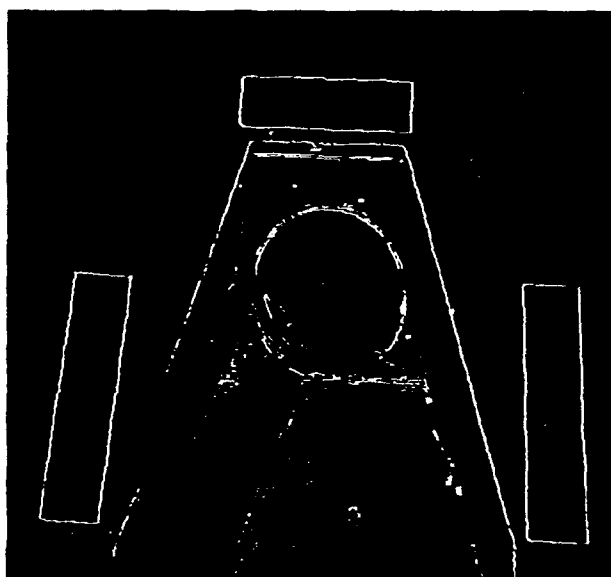


Figure 4.4. Non-Maximal Suppression Image With Thresholding.



hard coded threshold improved the overall performance of the edge detection, at the cost of a reduced range of lighting conditions that the edge detection functioned properly in. Figure 4.5 shows the non-maximum suppressed image in dark lighting without thresholding and Figure 4.6 illustrates the same lighting condition with thresholding (set at 3). Adaptive thresholding could be used to reduce the effect of lighting on image processing. Adaptive thresholding is the ability to adjust the threshold value based on the strength of the image pixels. Adaptive thresholding allows the system to adjust to environment. The gain in effective lighting range was not sufficient to warrant the impact of the adaptive threshold on processing time.

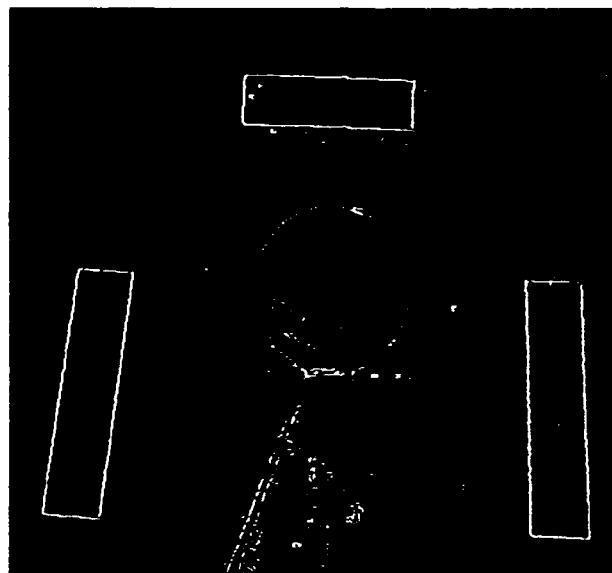


Figure 4.5. Non-Maximal Suppression Image Without Thresholding Dark Lighting.

### 4.3 Edge Processing

The edge processing was the second most time consuming stage of the BIPRS. The implementation time varied from 15 to 30 seconds depending on the number of positive pixels in the image. This stage includes building linked-lists of connected edge pixels, concatenation of edges together to form longer edges, and the formation of closed loops

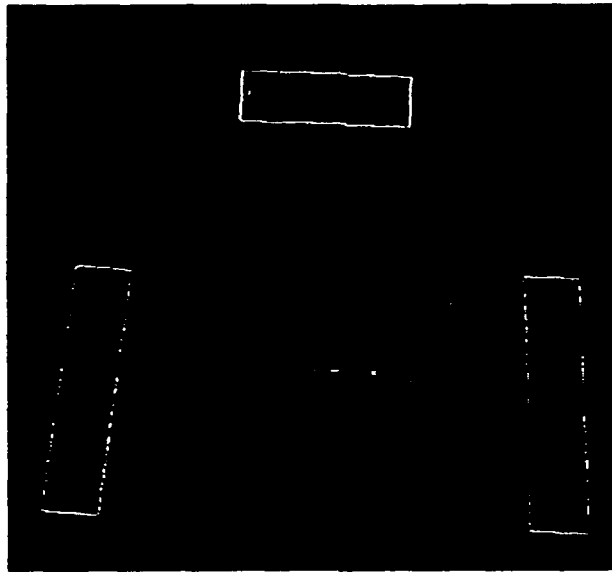


Figure 4.6. Non-Maximal Suppression Image With Thresholding Dark Lighting.

of connected pixels and is very sensitive to noise (stray pixels). Concatenation of edges is affected and the recognition process could fail if there are too many stray pixels in the image. This stage benefits the most from the thresholding conducted during the initial processing and is crucial for the success of the recognition process.

#### *4.4 Port Recognition*

The port recognition stage is the final step in the BIPRS and only requires about 15 seconds to complete. This stage contains all the logic to recognize the indicators. This stage examines the closed loops formed in the edge processing stage and looks for the attributes that identify the loops as the indicators around the port.

A number of tests were devised to evaluate the performance of the BIPRS. The most important attribute of the BIPRS was brightness invariance. After brightness invariance was tested, the invariance to rotation and size was evaluated. The last test was for port recognition with partial indicators visible in the image.

**4.4.1 Brightness Invariance** The BIPRS was able to identify the refueling port in a wide range of lighting conditions. The fluorescent lighting in the lab was not adjustable, so the iris of the CCD camera was adjusted to simulate variations in lighting. For dark conditions, the iris was closed to allow minimal light into the camera. For bright light conditions simulation, the iris of the camera was opened wide. The BIPRS performed well during a series of tests with a primer gray port surface with white rectangular indicators, that the refueling port was changed to a more challenging flat black with gray indicators, representative of new stealth aircraft. Figures 4.7 and 4.8 show the image acquisition and non-maximal suppression (NMS) processing under normal fluorescent room light conditions. Figures 4.9 and 4.10 illustrate the image processing under low light conditions (iris closed) and Figures 4.11 and 4.12 the image processing under bright light conditions (iris open). The BIPRS was able to pick out the rectangular indicators under all the different lighting conditions. The BIPRS did have some problems in very low lighting conditions (iris completely closed) picking out all three indicators. The edges of the three indicators were not well defined and further edge processing could not take place. When the iris was opened just 1/8 inch the BIPRS was able to identify the three indicators.

**4.4.2 Rotation Invariance** It is important to recognize the port when it is present in the image even if the port is at a bad refueling angle (angle where the refueling arm can't reasonably be inserted into the port). The recognition algorithm was able to identify the refueling port with the port rotated almost 5 degrees either side of center. Figure 4.13 shows the port rotated off center. When the port is off center by a large angle the recognition algorithm just identifies that the port is present at a bad angle and does not try to move to the port. The angle must be less than 10 degrees either side of the center line of the image to be at an acceptable angle to move the refueling arm into position. It would be difficult to determine the acceptability of the refueling angle to the port without the indicators.

**4.4.3 Size Invariance** The recognition algorithm is invariant to the size of the refueling port in the image. The BIPRS uses the ratio of length to width of the indicators to identify which indicators were found. An indicator with a length:width ratio of 4:1 at

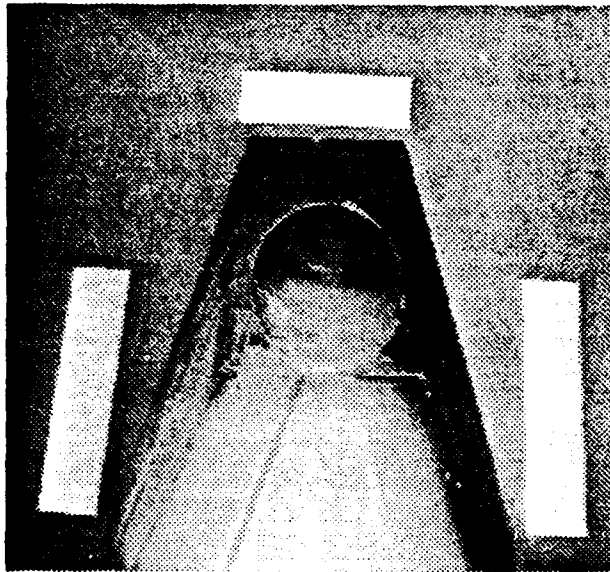


Figure 4.7. Original Image of the Refueling Port (Normal Room Light).

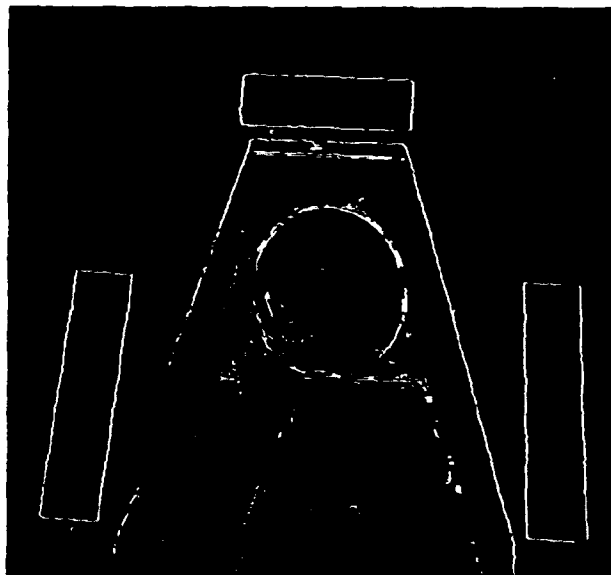
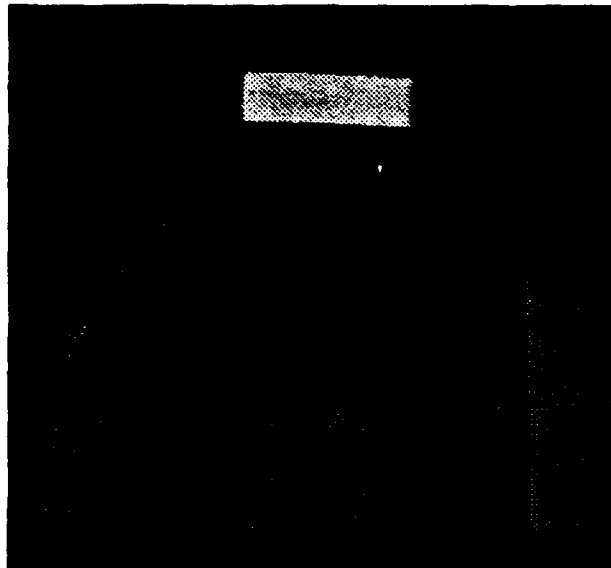
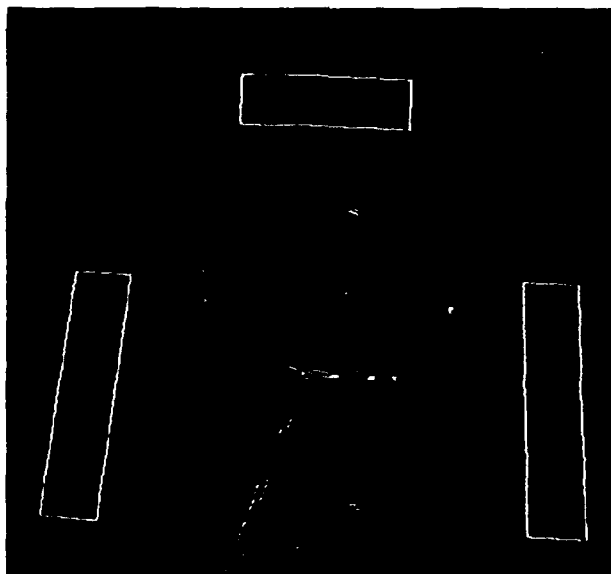


Figure 4.8. Non-Maximal Suppression Image of the Refueling Port Image (Normal Room Light).



**Figure 4.9. Image with Limited Light.**



**Figure 4.10. Non-Maximal Suppression of Image with Limited Light.**

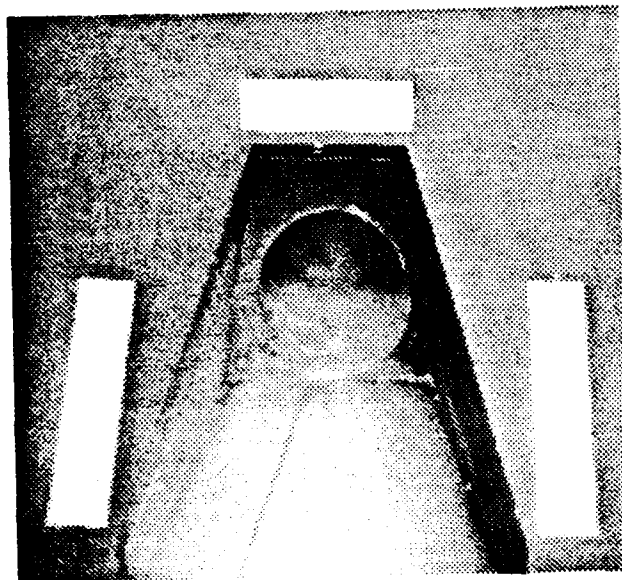


Figure 4.11. Image with Bright Light.

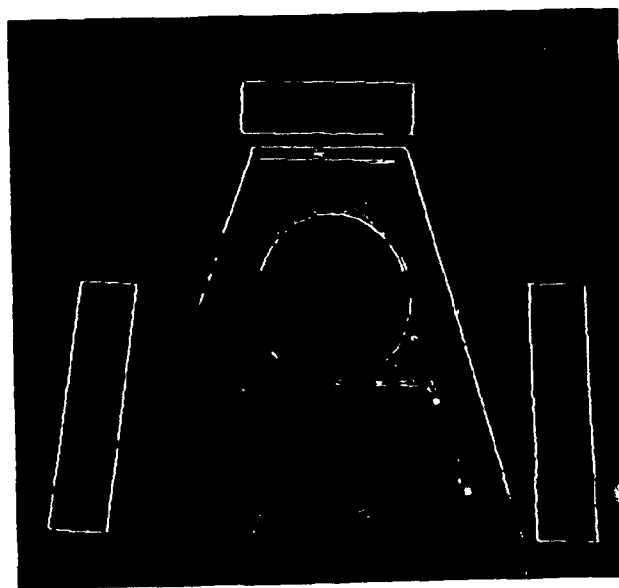


Figure 4.12. Non-Maximal Suppression Image with Bright Light.

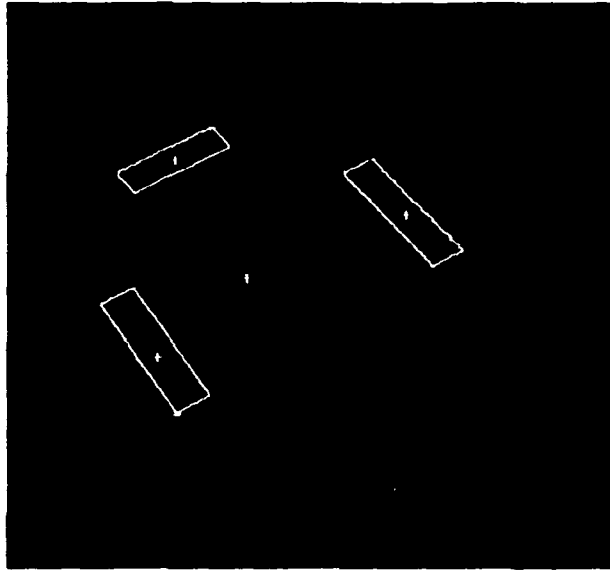


Figure 4.13. Rotated Image of the Refueling Port.

one meter from the base of the robot would have the same ratio of 4:1 at five meters or ten meters from the robot. Since the recognition algorithm is based on the shape of the indicators and the relationship of length vs width of each indicator, the recognition should be independent of the size of the port in the image as long as the indicators are visible. Size invariance was tested by first placing the port close to the robot and running the BIPRS and then moving the port away from the robot. Figure 4.14 shows the refueling port close in and Figure 4.15 with the port four meters from the base of the robot. If the port is located out of range of the robot, the port is centered in the image and the recognition process is repeated until the port is in range.

**4.4.4 Recognition with Partial Indicators** The BIPRS requires that all three indicators be completely visible before it sends position update information to the trajectory planner. If there are less than three full indicators visible, the BIPRS sends the position information to a routine to center the visible indicators in the image then take another image to find all the indicator. The recognition algorithm identified the refueling port when there were partial indicators visible. The port recognition criteria was relaxed when portions of

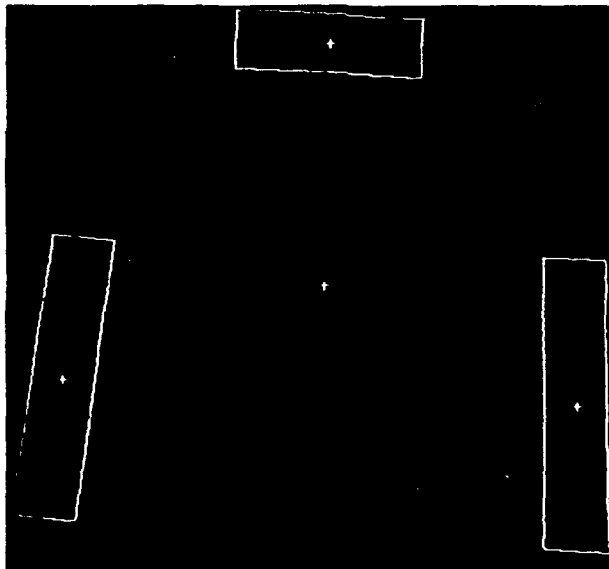


Figure 4.14. Image of the Refueling Port (Close).

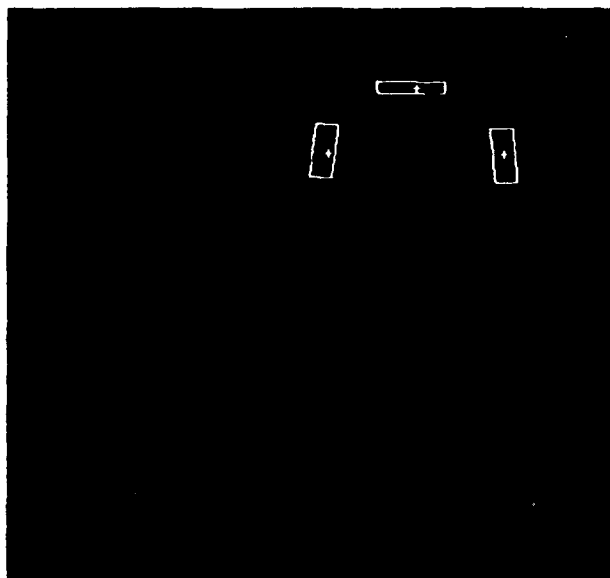


Figure 4.15. Image of the Refueling Port (Far).



the closed loops, that form the indicators were on the edge of the image. For example, if a possible side indicator was on the side of the image a ratio range of 4:1 to 8:1 would be used to identify the indicator. If there were partial indicators visible in the image, the center of mass for these indicators was calculated and the robot's position was sent to the centering routine to move the center of mass of the indicators to the center of the image. After the center of mass of the indicators is moved to the center of the image, the vision system takes another look for all three indicators. Figures 4.16 through 4.20 show different conditions where partial indicators were correctly identified. Figure 4.16 illustrates a complete top indicator and two partial lower indicators are visible. Figure 4.17 gives an example of the port with just the two side indicators visible. Figure 4.18 illustrates a partial top and one partial lower indicator visible. Figure 4.19 and 4.20 show a top indicator and one side indicator visible.

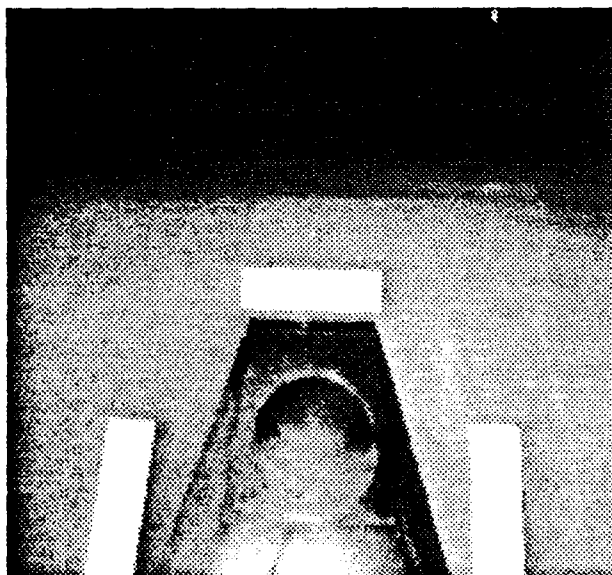
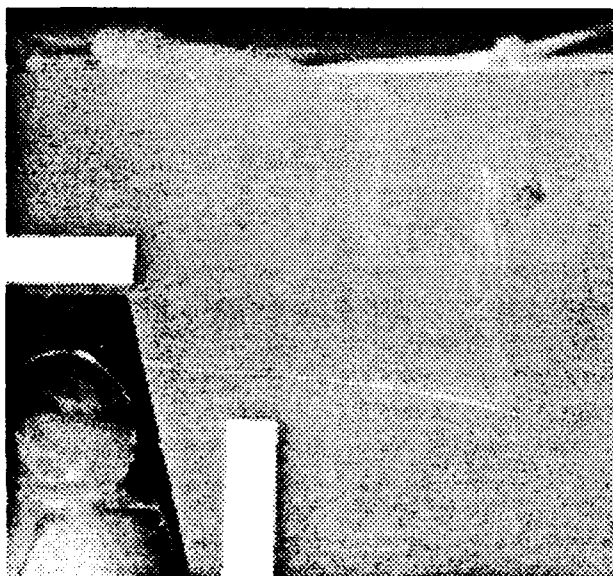


Figure 4.16. Top with Two Partial Indicators Present.



**Figure 4.17. Image with Two Indicators.**



**Figure 4.18. Partial Top and Side Indicators.**

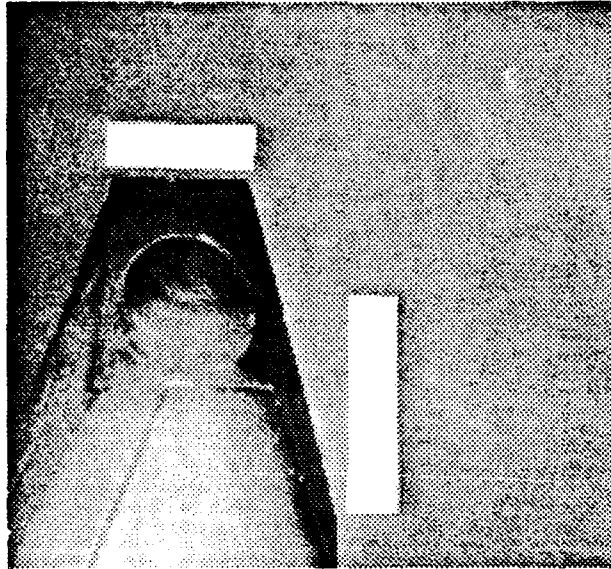


Figure 4.19. Top Indicator Plus Right Side Indicator.

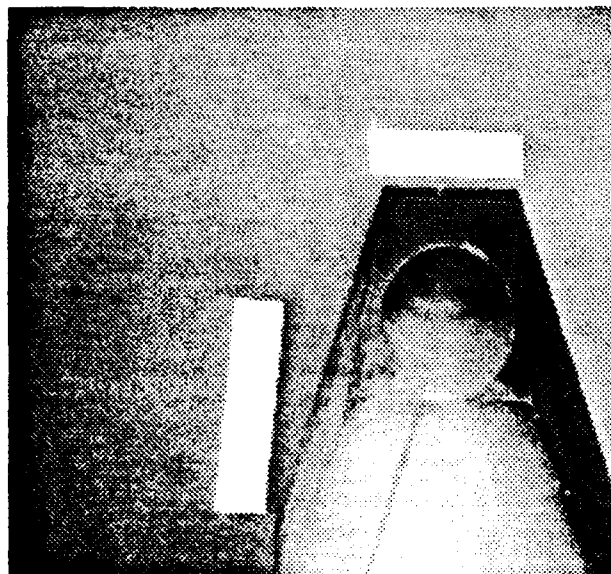


Figure 4.20. Top Indicator Plus Left Side Indicator.

#### 4.5 Searching for the Port

The search for the refueling port was accomplished autonomously. The robot was placed close to the refueling port at some initial position (see Figure 4.21) prior to beginning the search for the port. The search pattern assumes that the robot was placed close to the port but not necessarily directly at the port. Placing the robot close to the port was only for test purposes. The robot was able to start its search from any point in the 320 degree range of the PUMA robot and find the port, if the port is in the camera's field of view. The robot continued to search for the refueling port until it was identified. If the robot reached one of its mechanical stops during the search for the refueling port (identified by joint 1 at +160 or -160 degrees in the software), it reversed the direction of the search. If during the search for the port, at least two possible indicators are identified the robot will deviate from the search pattern to check for the port. If the port is not found the robot will resume the search. Figures 4.22 through 4.24 shows the process of homing in on the

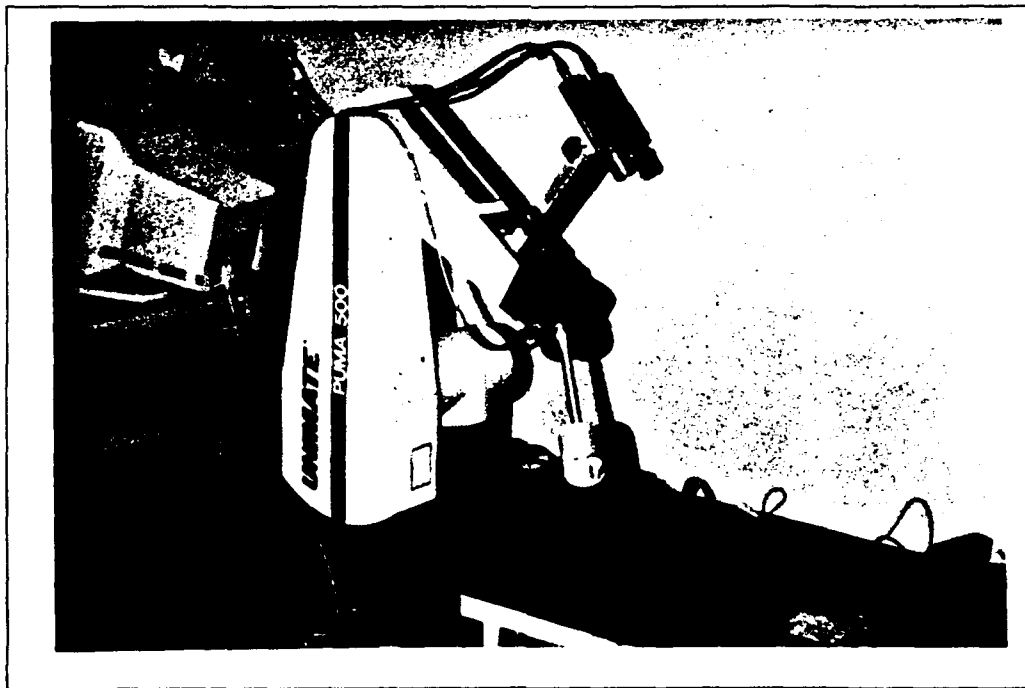


Figure 4.21. Initial Position for the Search for the Port.

refueling port.

In Figure 4.22 only two indicators are visible so the vision system departed from the



Figure 4.22. Search Image with Two Indicators Present.



Figure 4.23. Image with Two Indicators Centered.

search pattern and moved the two indicators towards the center of the image. Figure 4.23 shows that a third indicator was partially visible and the vision system indicated that there were three indicators, but since one was partially occluded, the vision system again centered the indicators. Figure 4.24 illustrates the next look where all the indicators are now visible, the port is recognized, and position update information is sent to the trajectory planner. Figure 4.25 shows the final image when all the indicators were identified and the center of mass calculated. Figures 4.26 and 4.27 highlight the final position of the PUMA robot when the port has been identified, was in range, and at a proper angle.

#### *4.6 Limitations and Problems*

One problem that was encountered during this research was with the Sobel Edge Operator (SEO) function on the ITEX image processing board. The ITEX SEO gives the wrong edge direction for one orientation of edges. The edge that should have a direction of zero ends up with a direction of two, off by 90 degrees. This edge directions error was noticed when trying to perform non-maximal suppression (NMS) of the image. NMS relies on the edge direction to determine which direction to compare neighbors to find the maximum point in the edge. With the wrong edge direction, the NMS function was comparing the wrong neighbors and destroying good edges. The problem was overcome by compensating for the error in the get direction, *Ddirect*, routine in the BIPRS software. The values will have to be corrected in the BIPRS software when the error in the ITEX software is corrected. The error on the ITEX board also affected the edge processing functions used in the BIPRS. The optimal implementation of an edge formation algorithm relies on the edge direction to determine the pixels to combine into edges. Since the edge direction information was unreliable a different approach had to be selected. The affects of the ITEX error rippled all the way through the BIPRS software. The problem with the ITEX information was overcome, but resulted in a less efficient software design for the BIPRS.

Camera misalignment also causes some problems when the refueling arm is moved to the refueling port. If the camera is not pointed directly at the end of the refueling nozzle, the BIPRS assumes that the refueling port is at a different location than it actually is. It is

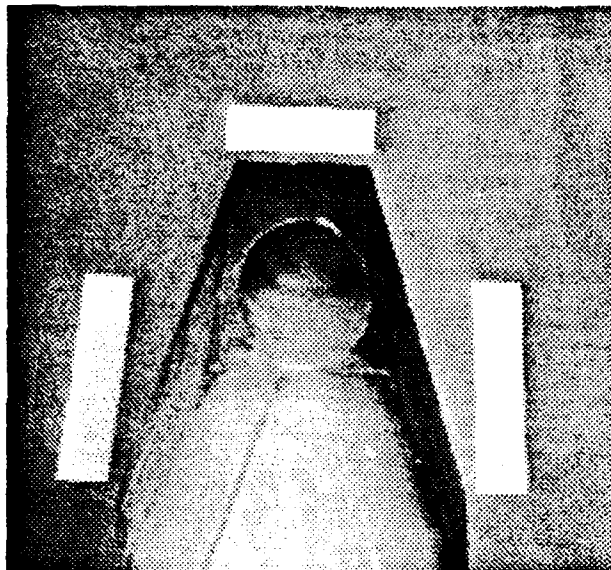


Figure 4.24. Final Image with All Indicators Visible.

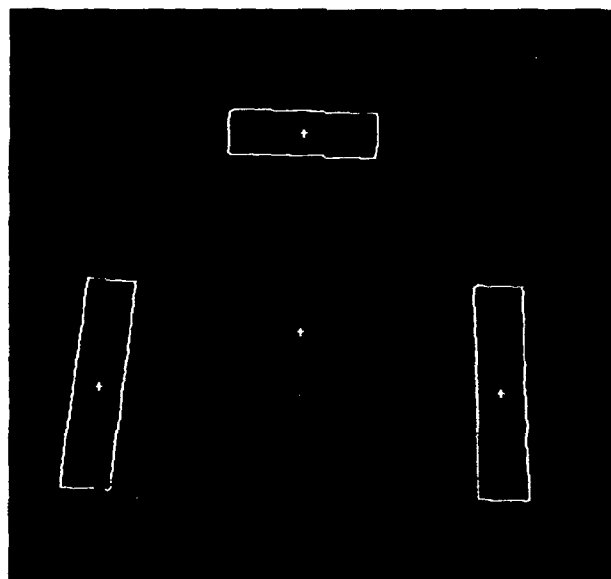


Figure 4.25. Final Indicator Image.

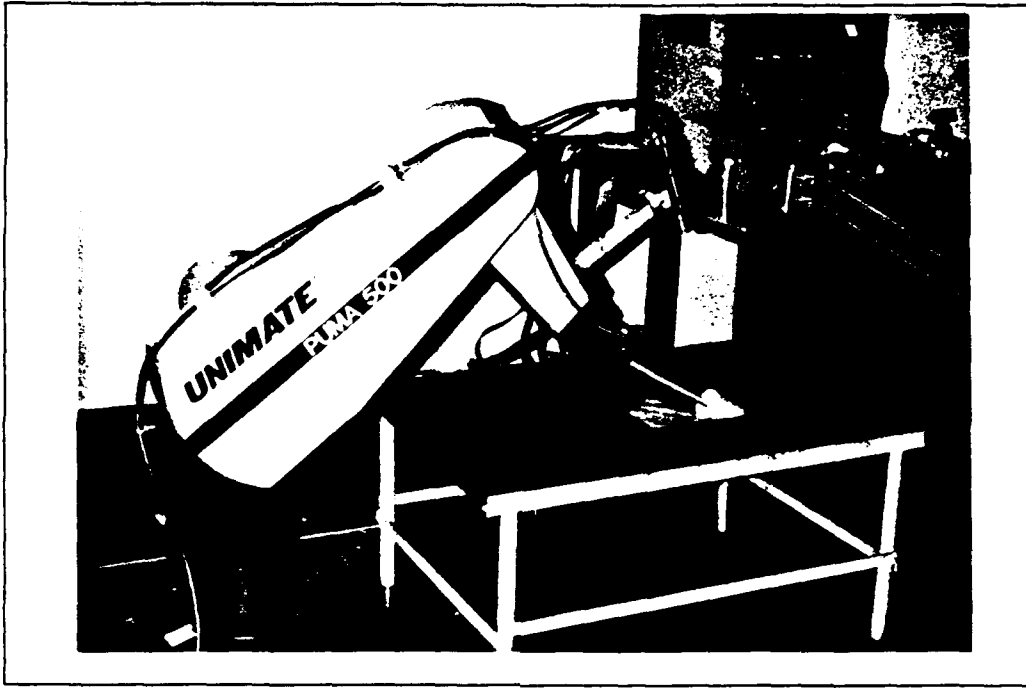


Figure 4.26. Final Position for Refueling.

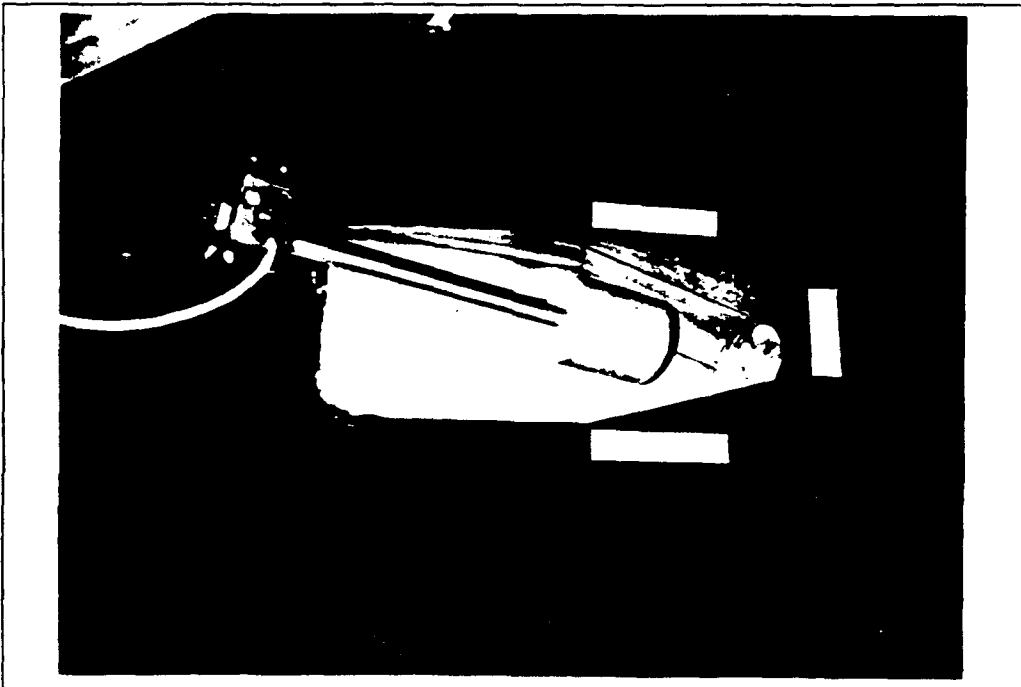


Figure 4.27. Closeup of Final Position for Refueling.



critical that the camera is properly aligned to prevent errors in positioning of the refueling nozzle.

#### *4.7 Conclusion*

The Brightness Invariant Port Recognition System (BIPRS) algorithm was able to locate and identify an aerial refueling port under a wide range of lighting conditions. The BIPRS is also position and rotations invariant. The improvement in the recognition algorithm that allows the detection of the refueling port with only half of the port visible in the image allows for a more efficient search strategy for the refueling port. One problem was encountered with the ITEX Sobel edge operator but it was overcome and had a minimum impact on the BIPRS implementation. The new algorithm provides AFIT with the needed recognition capability to test the autonomous robotic refueling under realistic unstructured lighting conditions.

## *V. Conclusions and Recommendations.*

### *5.1 Summary.*

A dynamic brightness, size, and rotation invariant port recognition algorithm was developed and demonstrated. The brightness invariant port recognition system (BIPRS) relied on edge detection to make the recognition algorithm invariant to lighting. The BIPRS was also able to recognize the refueling port when it was rotated as much as 45 degrees off center, and when the port was placed up to three meters from the robot. Another important capability that the BIPRS has demonstrated is recognizing the refueling port when it was only partially visible in the image. The brightness invariance was accomplished with edge detection, that looks for intensity changes in the image, and simple artificial intelligence (AI) algorithms. The rotation and size invariance, and partial port problems are also handle by simple AI algorithms that look for the identifiable attributes of the rectangular indicators in the image. The robot vision system consisted of a small Sony CCD camera mounted on the third link of a PUMA-560 robot arm and a VAXstation III computer. The BIPRS located and identified a half scale mock-up of an aircraft refueling port in a wide range of lighting conditions and supplied the port location to the trajectory planning routine. Existing robot servoing routines were used to command the robot arm along a trajectory which aligned the refueling nozzle with the port [22].

This research demonstrated that the basic concept of robotic refueling of aircraft was feasible. The basic concept being researched was for the BIPRS to locate and identify the aircraft refueling port and provide the information to a trajectory planner to adjust the position of the robot arm. The robot carrying the refueling nozzle is then moved into position close to the port. After the robot was moved to the port compliant motion techniques are used to perform the actual insertion of the refueling nozzle. The BIPRS used Sobel edge detection to find and isolate three rectangular indicators located around the refueling. The BIPRS was able to locate the refueling port in a wide range of lighting conditions. The BIPRS was also able to identify the refueling port when it was rotated as much as 45 degrees away from the camera and up to three meters from the base of

the robot. The current BIPRS algorithm is too slow to be used operationally, but has validated the basic concept of visual servoing for coarsely positioning the refueler robot.

*5.1.1 Port Recognition.* The experimental results confirmed that edge detection is an effective means of isolating the indicators from the rest of the objects in the image. Shipman's rectangular indicators provided perfect long straight edges to perform edge detection on. The rectangular indicators simplify the refueling port recognition process by providing easily recognizable shapes. The width of the top indicator provided an accurate estimate of the range to the refueling port. With the range information provided by the top indicator, the robot arm was accurately placed above the refueling port.

*5.1.2 Light Invariance.* Edge detection provided a vision system that work well under a wide range of lighting conditions. The edge detection algorithm had some problems identifying all three indicators, under limited light conditions when the threshold was set at three. When the threshold was reduced to zero or one the algorithm was more effective in identifying the indicators in the dark range but had problems with noise in normal fluorescent room light conditions.

*5.1.3 Partial Indicators.* Using simple AI techniques and knowledge driven recognition algorithms the BIPRS was able to identify the refueling port with only part of the port visible. A more efficient search strategy was implemented to locate the refueling port by taking advantage of the partial indicator recognition capability provided by the BIPRS.

## *5.2 Conclusion.*

The BIPRS demonstrated that the concept of using a computer vision system to locate the refueling port is feasible. The vision system was able to identify the refueling receptacle in different lighting conditions and in different orientations. The compliant motion research plus the BIPRS demonstrate that robotic aircraft refueling can be accomplished and provides an incentive for additional research in autonomous robot applications at AFIT.

### 5.3 Recommendations.

**5.3.1 Thresholding.** The thresholding function needs to adapt to the light intensity in the image. The current thresholding algorithm removes all pixels that are below the threshold from further processing. At low light intensities many of the edge pixels become too weak and are deleted. An adaptive thresholding scheme would allow the recognition algorithm to adjust the threshold value used to remove noise based on the pixel strengths. One possible approach is to provide a light sensor whose measures of the light intensity present could be used to adapt the threshold value.

**5.3.2 Vaccelerator Board Usage.** The current vision algorithms needs to be implemented on the Vaccelerator board to speed up the image acquisition time and decrease the port recognition process time. The Vaccelerator board is a Q-BUS DMA peripheral containing its own processor, floating point hardware, and 4-megabyte memory, manufacture by Avalon [2]. The use of the Vaccelerator board can significantly reduce the processing time for the BIPRS.

**5.3.3 ITEX Image Processing Board.** The Sobel edge operator (SEO) function on the ITEX board needs to be corrected to provide proper edge direction for edges in the image.

**5.3.4 Edge Processing.** The edge processing algorithm needs to be improved and optimized to enhance its performance. The current edge processing algorithm is inefficient and could connect wrong edges together if there are two close proximity edges. This is partially due to the SEO problem identified above.

**5.3.5 Camera Calibration.** A method needs to be identified and implemented to calibrate the camera pointing angle to prevent camera pointing errors from affecting the BIPRS location accuracy.

**5.3.6 Dynamic Tracking.** Dynamic tracking software needs to be developed to allow the vision system to lock on and track a moving refueling port.

## *Appendix A. Brightness Invariant Port Recognition System (BIPRS) Software Flow*

### *A.1 Introduction*

The purpose of this appendix is to provide a basic understanding of the logic flow of the software developed for the Brightness Invariant Port Recognition System (BIPRS). Figure A.1 provides the flow of control for the initial start up of the BIPRS and the figures that appear after Figure A.1 present the logic flow for the major subroutines of the BIPRS. All the code for this research effort was written in the C programming language. The actual code written for this effort can be found in the AFIT Robotics Systems Laboratory in an internal report (ARSL-90-8) [18].

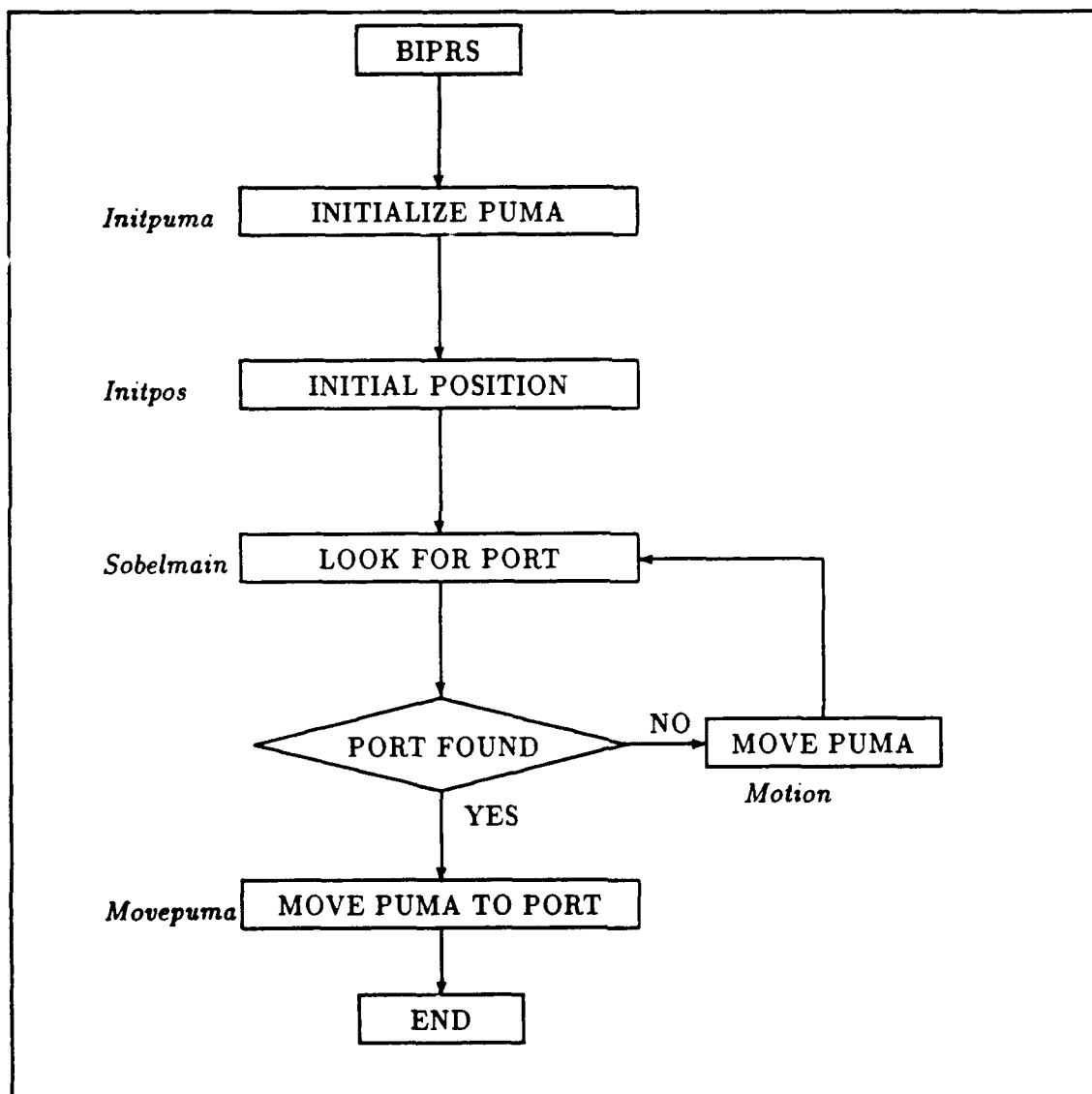


Figure A.1. Brightness Invariant Port Recognition System (BIPRS) Logic Flow

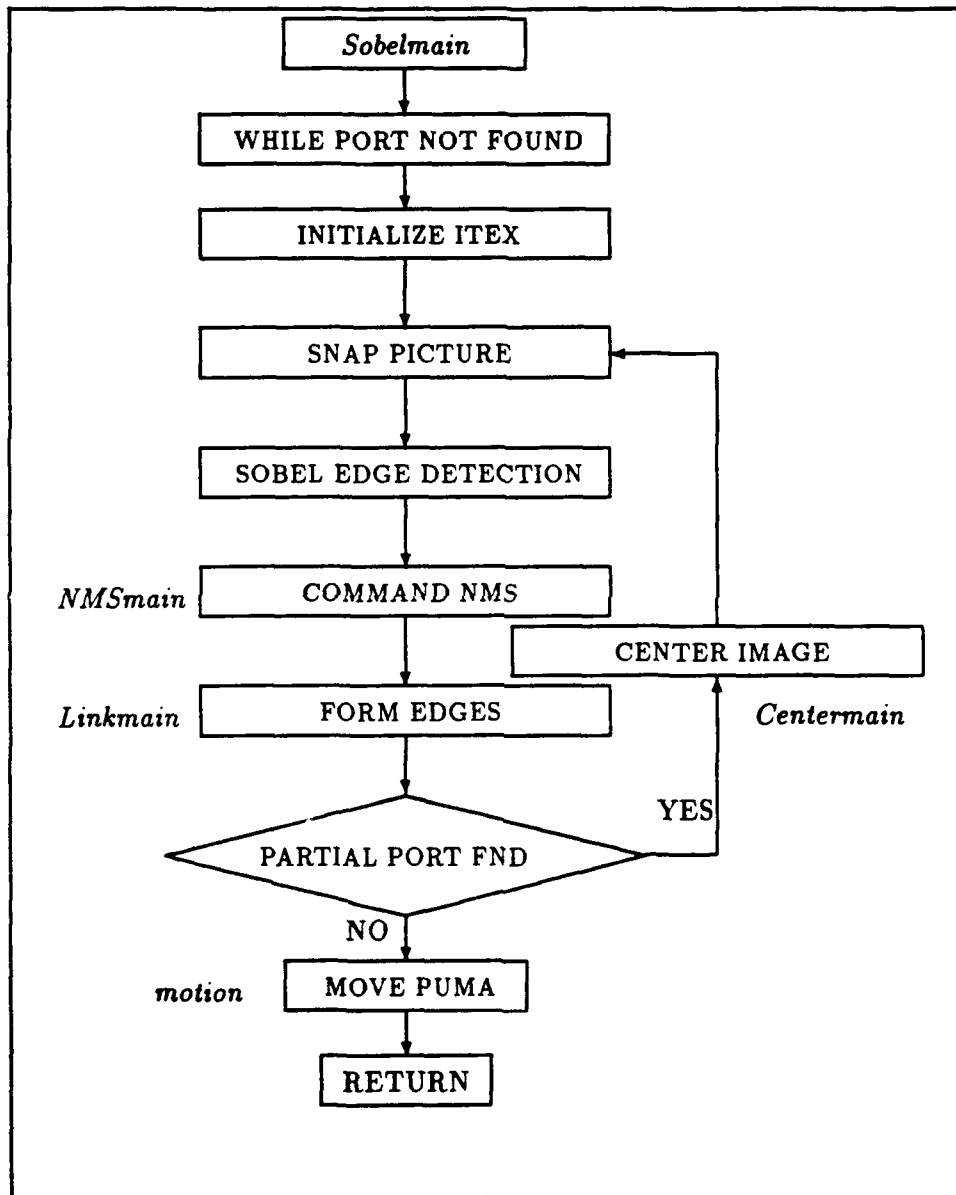


Figure A.2. *Sobelmain* Logic Flow

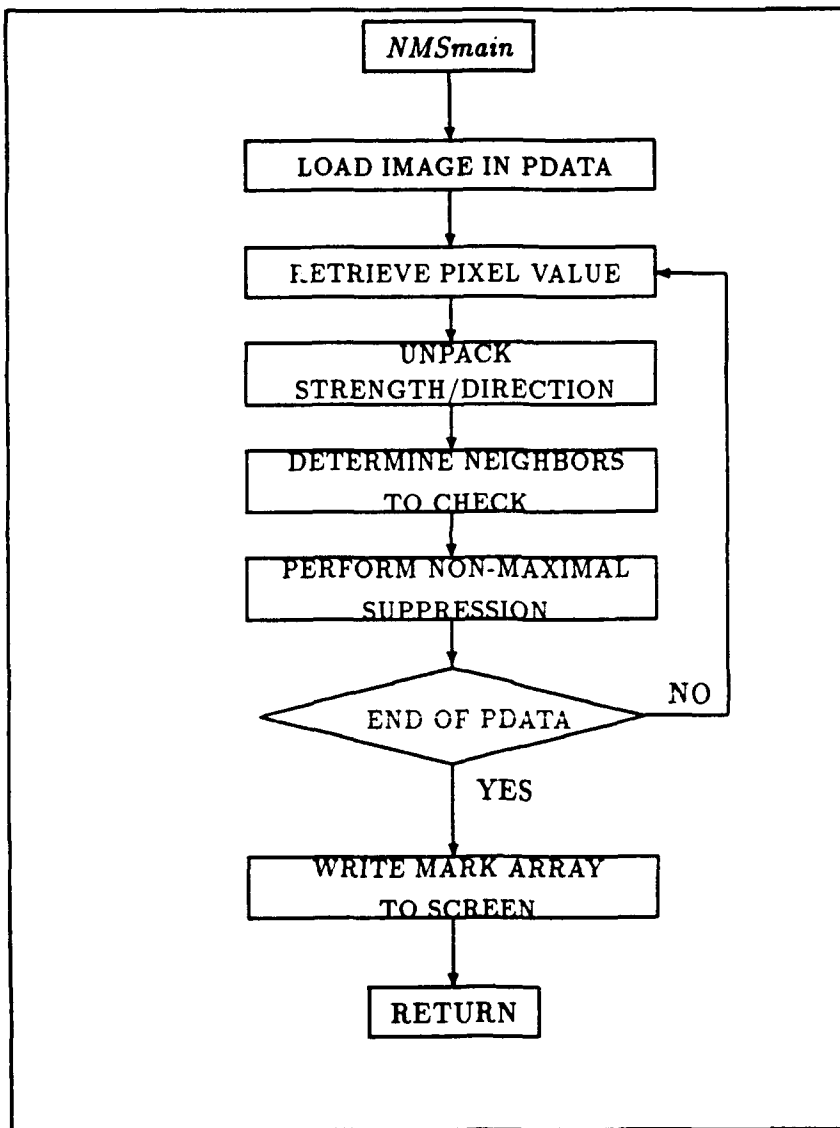


Figure A.3. *NMSmain* Logic Flow



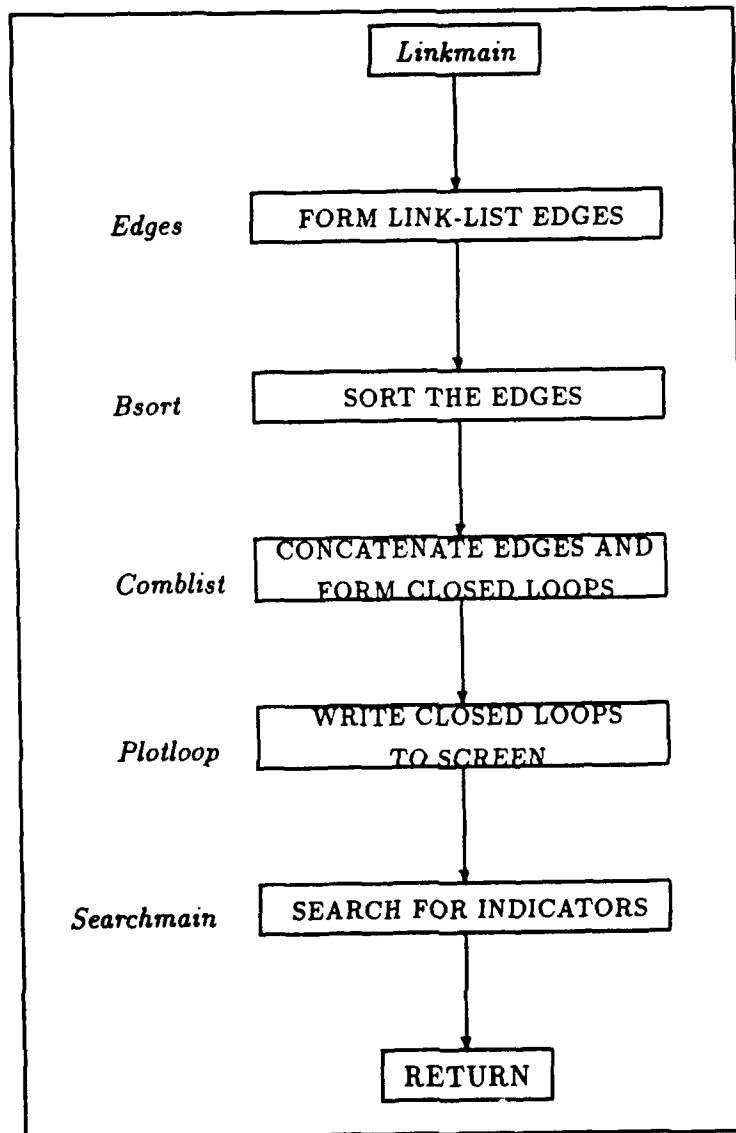


Figure A.4. *Linkmain* Logic Flow

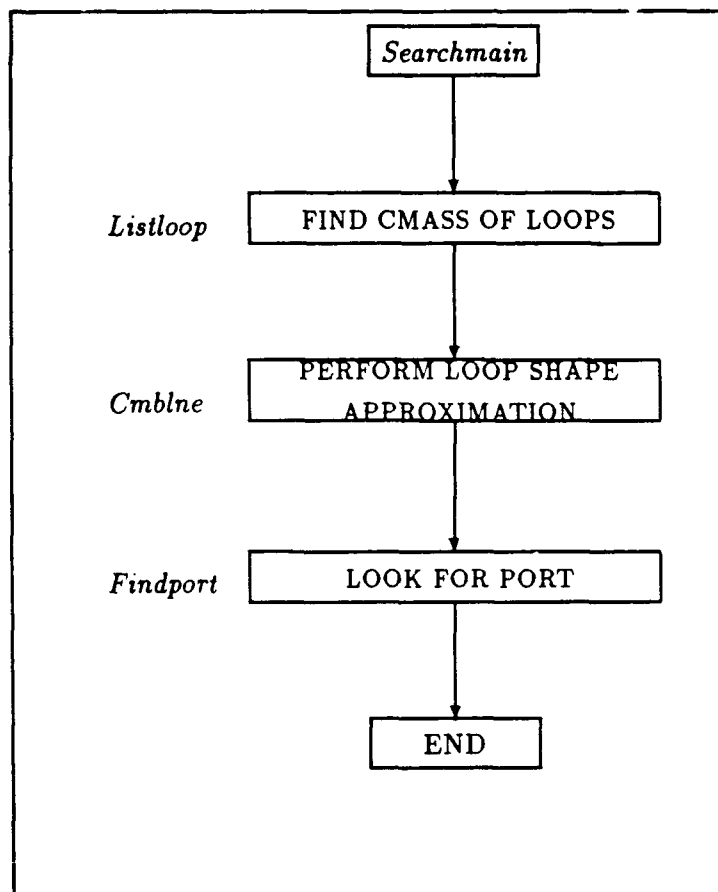


Figure A.5. *Searchmain* Logic Flow

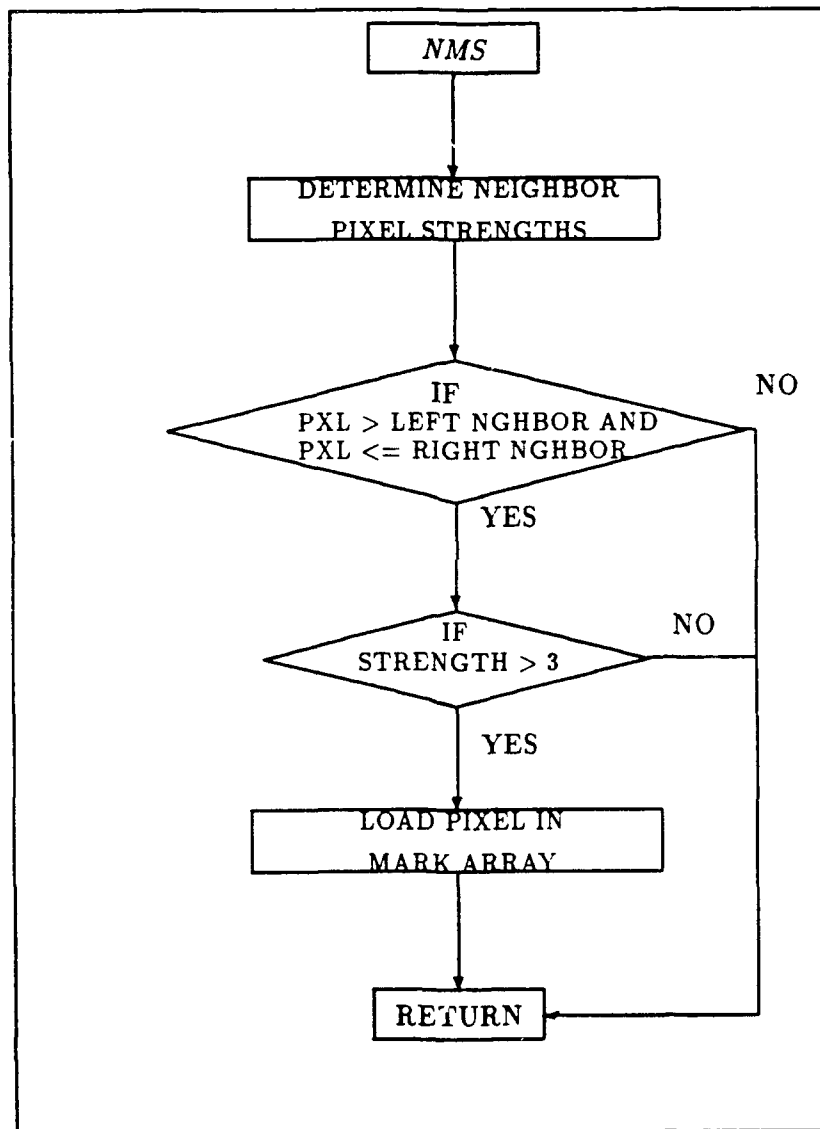


Figure A.6. Non-Maximal Suppression Logic Diagram

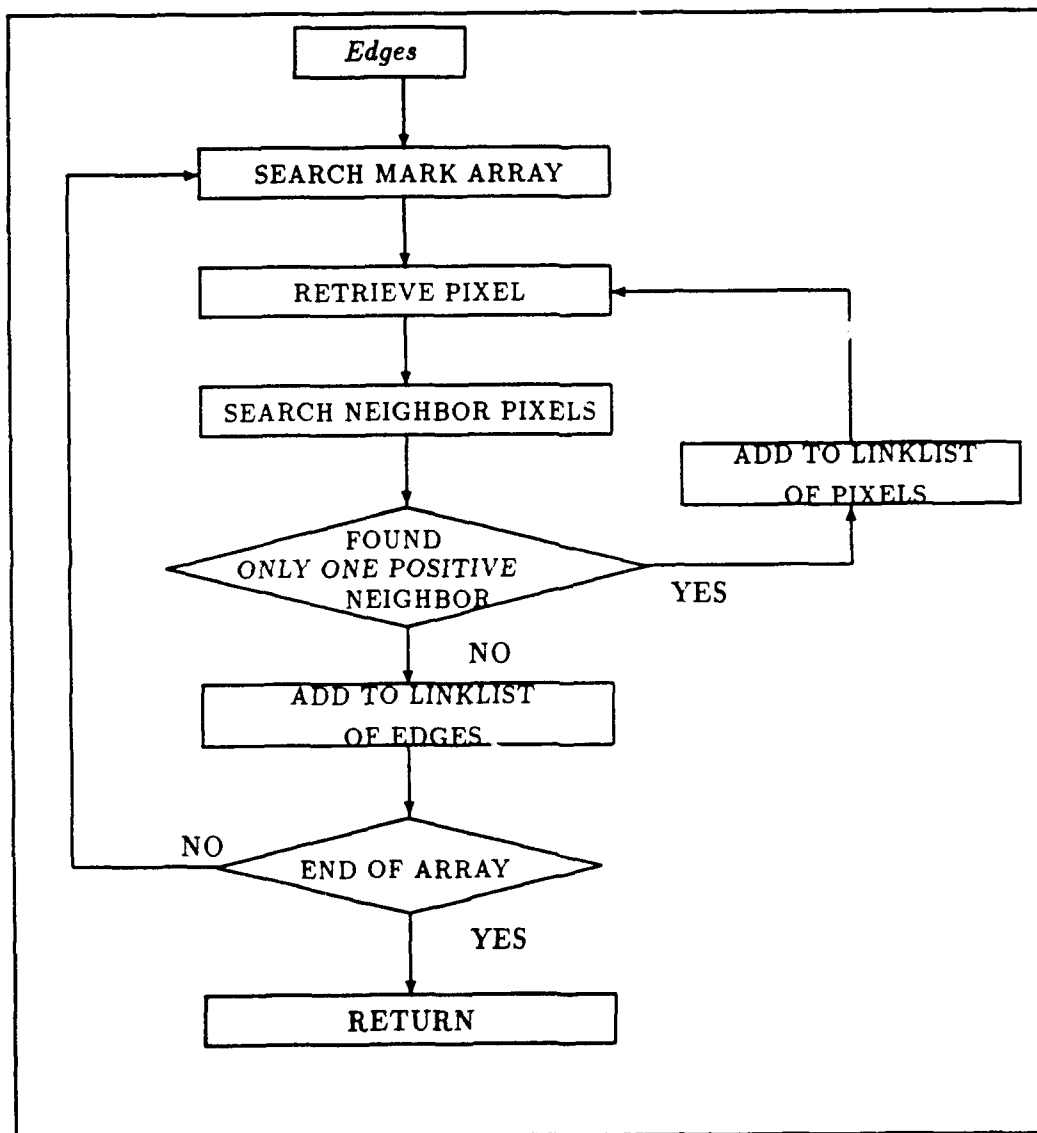


Figure A.7. *Edges* Logic Diagram

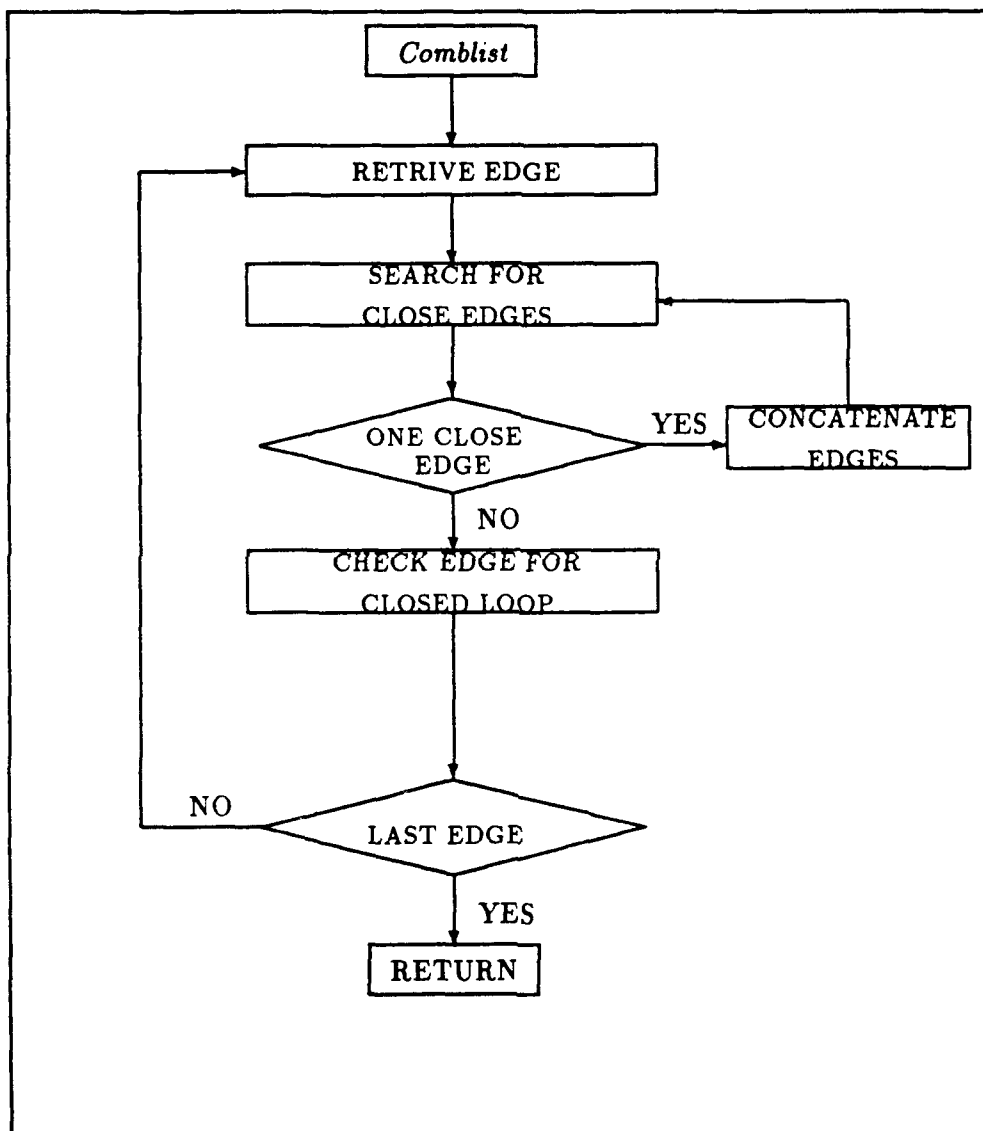


Figure A.8. *Comlist* Logic Diagram

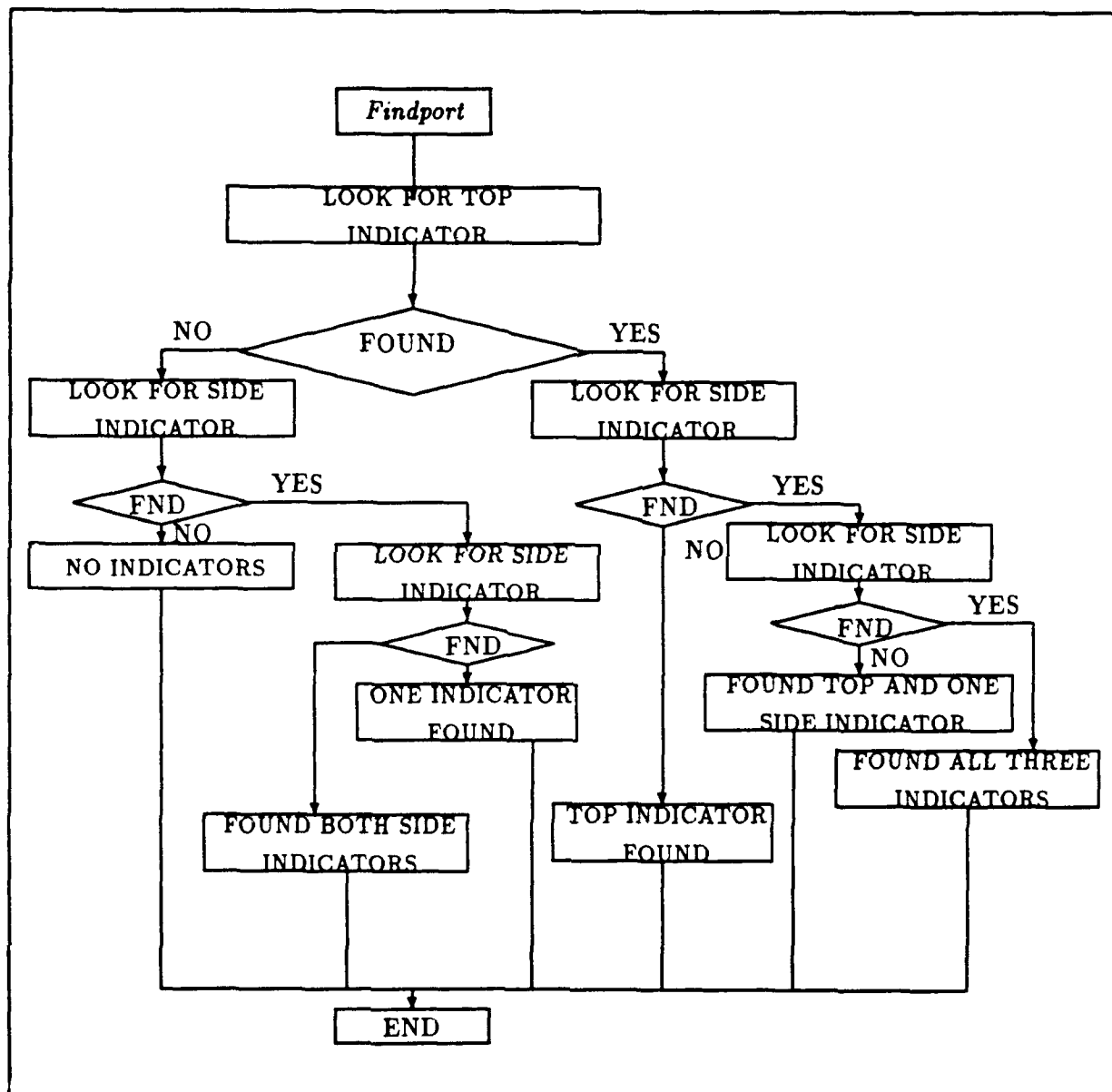


Figure A.9. Findport Logic Diagram

## Bibliography

1. Andersen, C. "Three Degree of Freedom Compliant Motion Control for Robotic Aircraft Refueling". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1990. AFIT/GAE/ENG/90D-01.
2. Inc. Avalon Computer System. "Vaccelerator Operation's Manual". Glendale, California, June 1989.
3. Shiu Y. C. "Grouping Image Features Into Loops for Monocular Recognition". In *Proc IEEE Int Conf on Sys, Man, and Cybernetics*, pages 752-762, November 1989.
4. Shiu Y. C. Edge detection and loop detection, June 1990. Personal Communications.
5. Ballard D. and C. Brown. *Computer Vision*. Prentice Hall, 1982.
6. Hecht E. and A. Zajac. *Optics*. Addison-Wesley, 1979.
7. Wahl F. *Digital Image Signal Processing*. Artech House, 1987.
8. Tan H. L. and S. B. Gelfand and E. Delp. "A Comparative Cost Function Approach to Edge Detection". *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1337-1349, Nov/Dec 1989.
9. Inc. Imaging Technology. "FG-100-Q User's Manual.". Woburn, Massachusetts, July 1987.
10. Eom K. and R. L. Kashyap. "Composite Edge Detection with Random Field Models". *IEEE Trans. on Systems, Man, and Cybernetics*, 20(1):81-93, Jan/Feb 1990.
11. Hertz L. and R. Schafer. "Multilevel Thresholding Using Edge Matching". *Computer Vision and Image Processing*, 44:279-295, 1986.
12. Lambert, L. "Evaluation and Enhancement of the AFIT Autonomous Face Recognizer Machine". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1987. AFIT/GE/ENG/87D-35 (AD-A188819).
13. Abidi M. and R. Gonzalez. "The Use of Multisensor Data for Robotic Applications". *IEEE Trans. on Robotics and Automation*, 6(2):159-174, April 1990.
14. Leahy M. "Robot Applications", August 1989. Class Handout Distributed in EENG540, Robotic Fundamentals, School of Engineering, Air Force Inst. of Tech. Wright-Patterson AFB, OH.
15. Leahy M., V. Milholen, and R. Shipman. "Robotic Aircraft Refueling: A Concept Demonstration". In *Proc of NAECON*, pages 1145-1150, May 1990.
16. Miller, M. "Implementation of a Visual Servoing System for Evaluation of Robotic Refueling Applications.". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1987. AFIT/GE/ENG/87D-45, (AD-A189676).
17. Jayant N. and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Vidio*. Prentice Hall, 1984.

18. Bennett R. "Brightness Invariant Recognition Algorithm Software", December 1990. ARSL-90-8, Air Force Inst. of Tech.
19. Schalkoff R. *Digital Image Processing and Computer Vision*. Wiley and Sons, 1989.
20. Robb, B. "Autonomous Face Recognition Machine Using a Fourier Feature Set". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1989. AFIT/GE/ENG/89D-44, (AD-A216040).
21. Ruck, D. "Multisensor Target Detection and Classification". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1987. AFIT/GE/ENG/87D-56.
22. Shipman, R. "Visual Servoing for Autonomous Aircraft Refueling". Master's thesis, School of Engineering, Air Force Inst. of Tech., Wright-Patterson AFB, OH, December 1989. AFIT/GE/ENG/89D-48, (AD-A216042).
23. Martin T., W. Kobel, S. Rogers, M. Kabrisky, and J. Mills. "A Distortion-Invariant Pattern Recognition Algorithm". In *Proc. SPIE Conf. Intelligent Robot and Computer Vision*, volume 726, pages 63-72, 1986.
24. Nalwa V. and T. Binford. "On Detecting Edges". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(6):699-714, November 1986.
25. Torre V. and T. Poggio. "On Edge Detection". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(2):147-163, March 1986.



### *Vita*

Richard Bennett was born on June 15, 1953 in Niagara Falls, New York. He spent most of his childhood in Southern California. He graduated from high school in Norwalk, California in 1971. In 1972, he enlisted in the Air Force and attended the Defense Language Institute (DLI) in Monterey California and became a Bulgarian Linguist. In 1973 he married Kathleen Ivens. He graduated from DLI in 1974 and was assigned to the National Security Agency, Fort Meade, Maryland. In 1981, He was accepted for the Airman's Education and Commissioning Program. He attended Texas Tech University, Lubbock, Texas and graduated in 1983 with a Bachelor of Science in Electrical Engineering. After graduation, he attended the Air Force Officer's Training School in San Antonio, Texas and on September 9, 1983 he was commissioned a Second Lieutenant.

His first duty assignment was at Nellis Air Force, Las Vegas, Nevada and he was assigned to Precision Location Strike System (PLSS) Initial Operational Test and Evaluation Test Team as a software evaluator. In 1987, he was reassigned to an Electronic Security Command detachment at Nellis Air Force Base where he worked as the Chief of the Engineering Branch. In 1989, he was assigned to the Air Force Institute of Technology (AFIT) School of Engineering, Wright-Patterson AFB, Ohio to earn a Master's Degree in Electrical Engineering.

Permanent address: 1336 Dequincy Dr., Xenia  
Ohio 45385

REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302 and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1990		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE BRIGHTNESS INVARIANT PORT RECOGNITION FOR ROBOTIC AIRCRAFT REFUELING			5. FUNDING NUMBERS	
6. AUTHOR(S) Richard A. Bennett, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/90D-04	
9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES) WRDC/FIVMB WPAFB, OH 45433-6583			10. SPONSORING MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>The purpose of this thesis effort was to develop and test a brightness invariant recognition algorithm that would locate and identify the half-scale mock-up of a Universal Aerial Refueling Receptacle Slipway Installation (UARRSI) aerial refueling port in different lighting conditions. This was accomplished by developing a brightness invariant port recognition system which relied on edge detection, line merging, loop formation, and knowledge driven recognition of identifiable attributes of the refueling port to locate and identify the UARRSI port. Based on this approach, a brightness invariant port recognition system (BIPRS) was demonstrated which identified the refueling receptacle in different lighting conditions. The BIPRS was also invariant to orientation and size of the UARRSI port. The BIPRS demonstrated the feasibility of the autonomous aircraft refueling task, and provides an incentive for additional research in autonomous robot applications. &lt;</p>				
14. SUBJECT TERMS Brightness invariance, edge detection, line merging, knowledge driven recognition, size invariance, and rotation invariance			15. NUMBER OF PAGES 88	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	